S01P1116

# FIG. 1

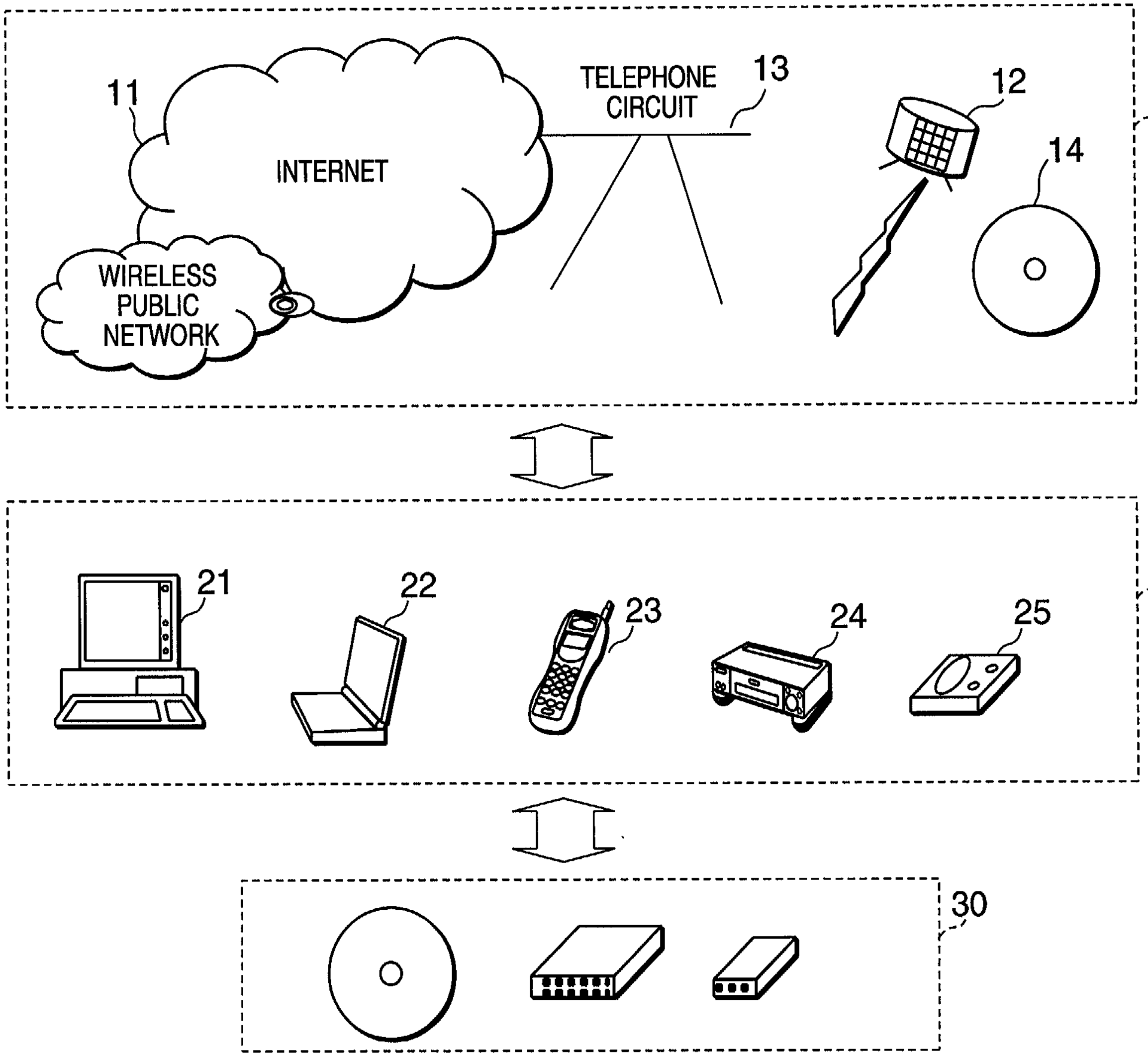
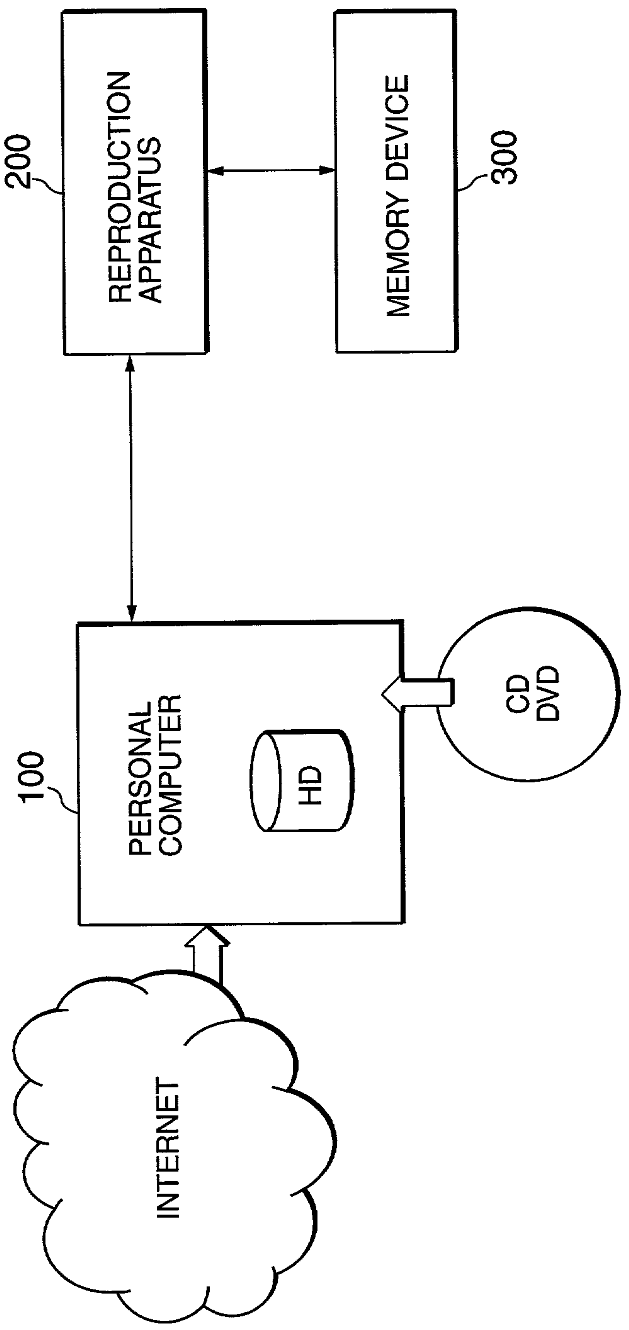
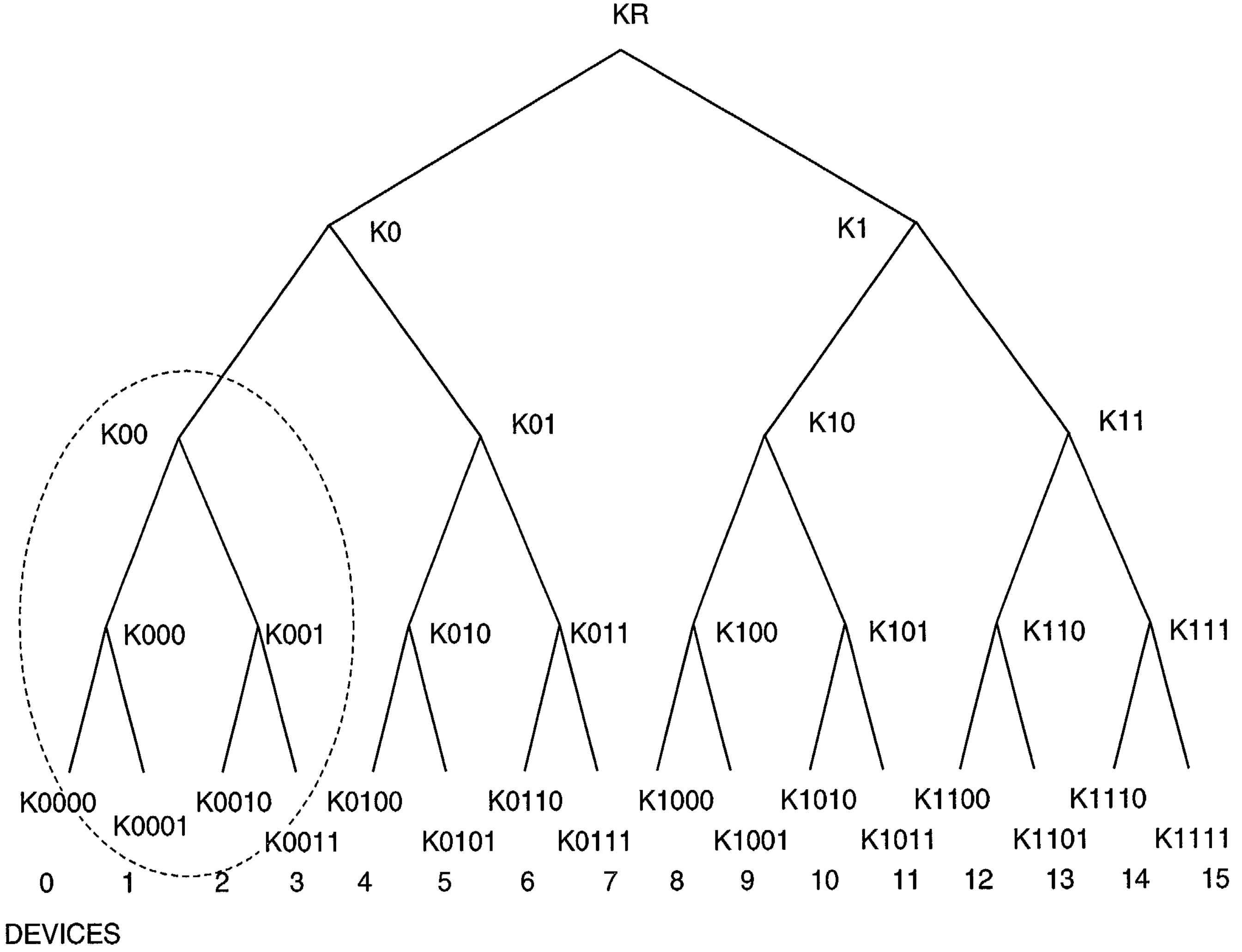


FIG. 2


100
PERSONAL COMPUTER
HD
CD DVD
INTERNET
200
REPRODUCTION APPARATUS
MEMORY DEVICE
300

# FIG. 3


KR
K0
K1
K00
K01
K10
K11
K000
K001
K010
K011
K100
K101
K110
K111
K0000
K0001
K0010
K0011
K0100
K0101
K0110
K0111
K1000
K1001
K1010
K1011
K1100
K1101
K1110
K1111
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
DEVICES

# FIG. 4

(A)

EKB (ENABLING KEY BLOCK) EXAMPLE 1
DELIVERS NODE KEYS OF VERSION (t) TO DEVICES 0, 1,AND 2

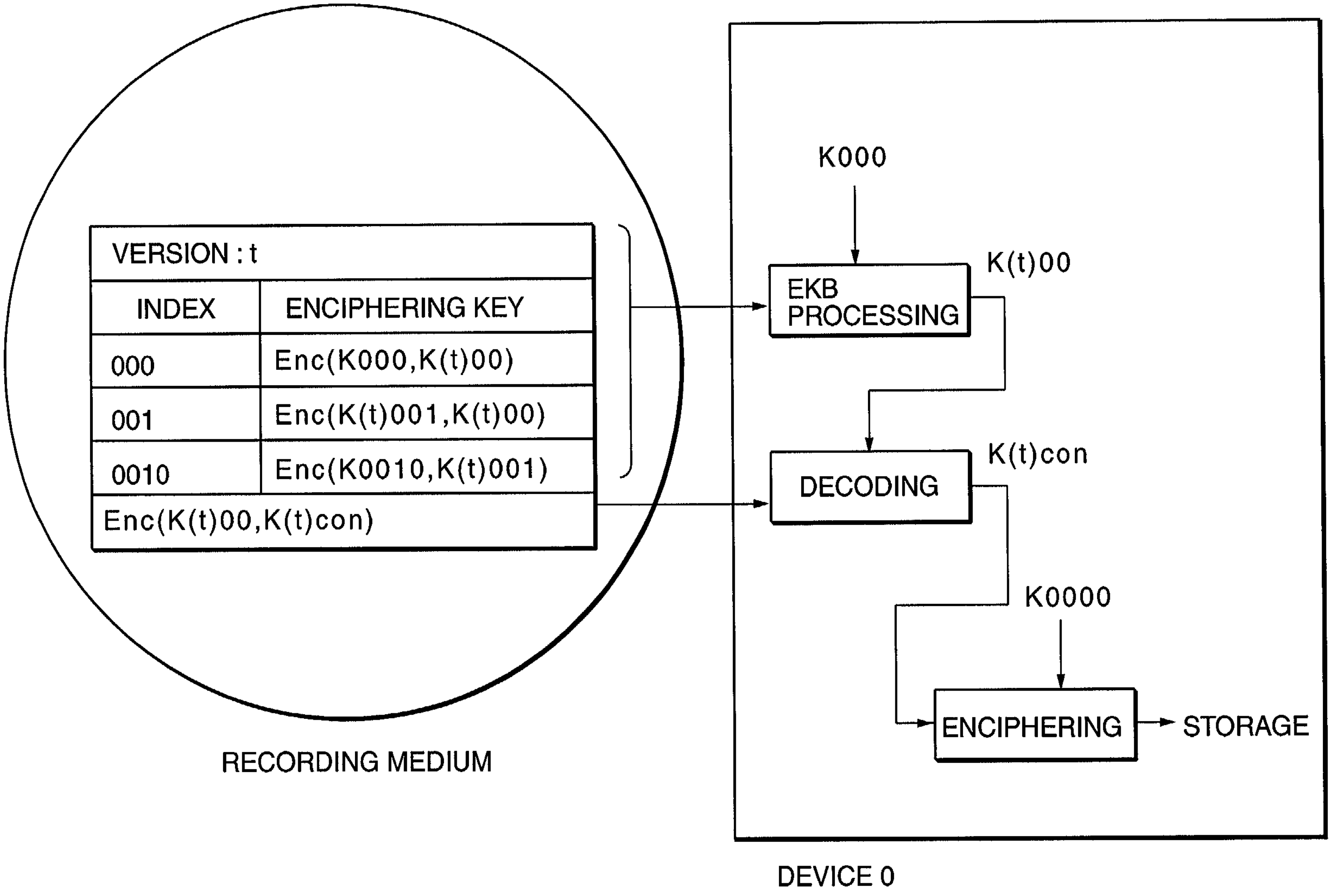
| VERSION : t | |
|---|---|
| INDEX | ENCIPHERING KEY |
| 0 | Enc(K(t)0, K(t)R) |
| 00 | Enc(K(t)00, K(t)0) |
| 000 | Enc(K000, K(t)00) |
| 001 | Enc(K(t)001, K(t)00) |
| 0010 | Enc(K0010, K(t)001) |

(B)

EKB (ENABLING KEY BLOCK) EXAMPLE 2
DELIVER NODE KEY OF VERSION (t) TO DEVICES 0, 1, AND 2

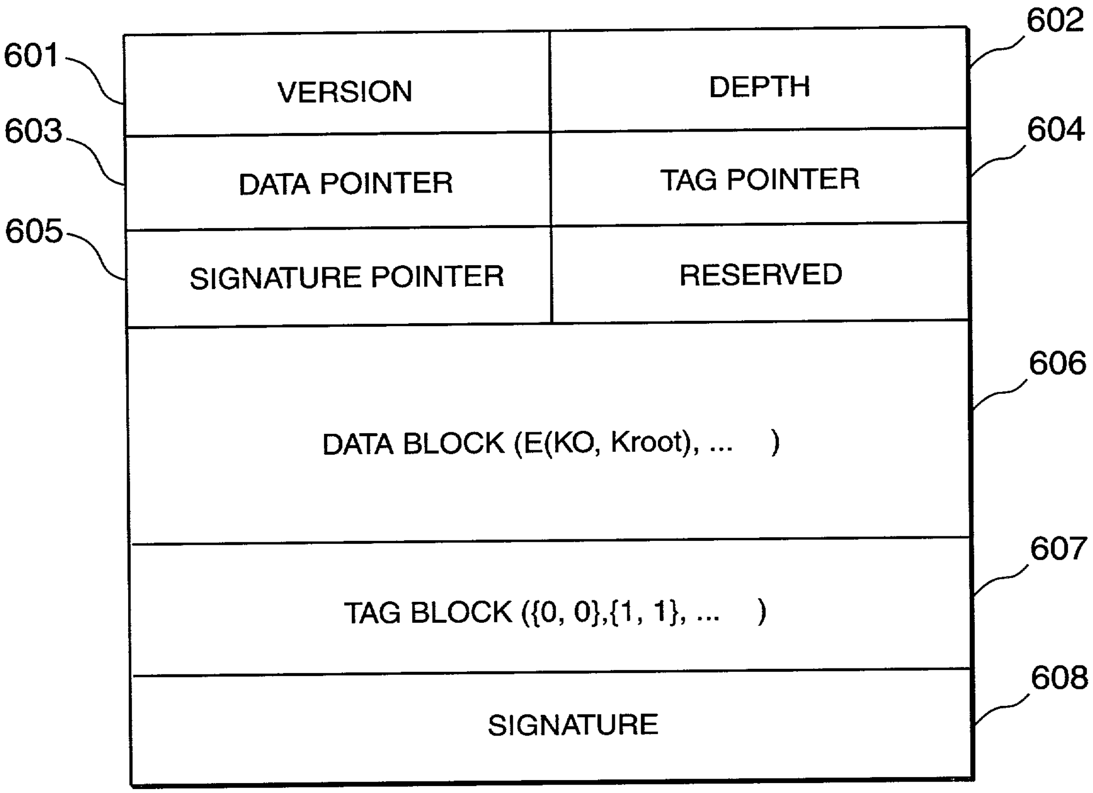
| VERSION : t | |
|---|---|
| INDEX | ENCIPHERING KEY |
| 000 | Enc(K000, K(t)00) |
| 001 | Enc(K(t)001, K(t)00) |
| 0010 | Enc(K0010, K(t)001) |

# FIG. 5

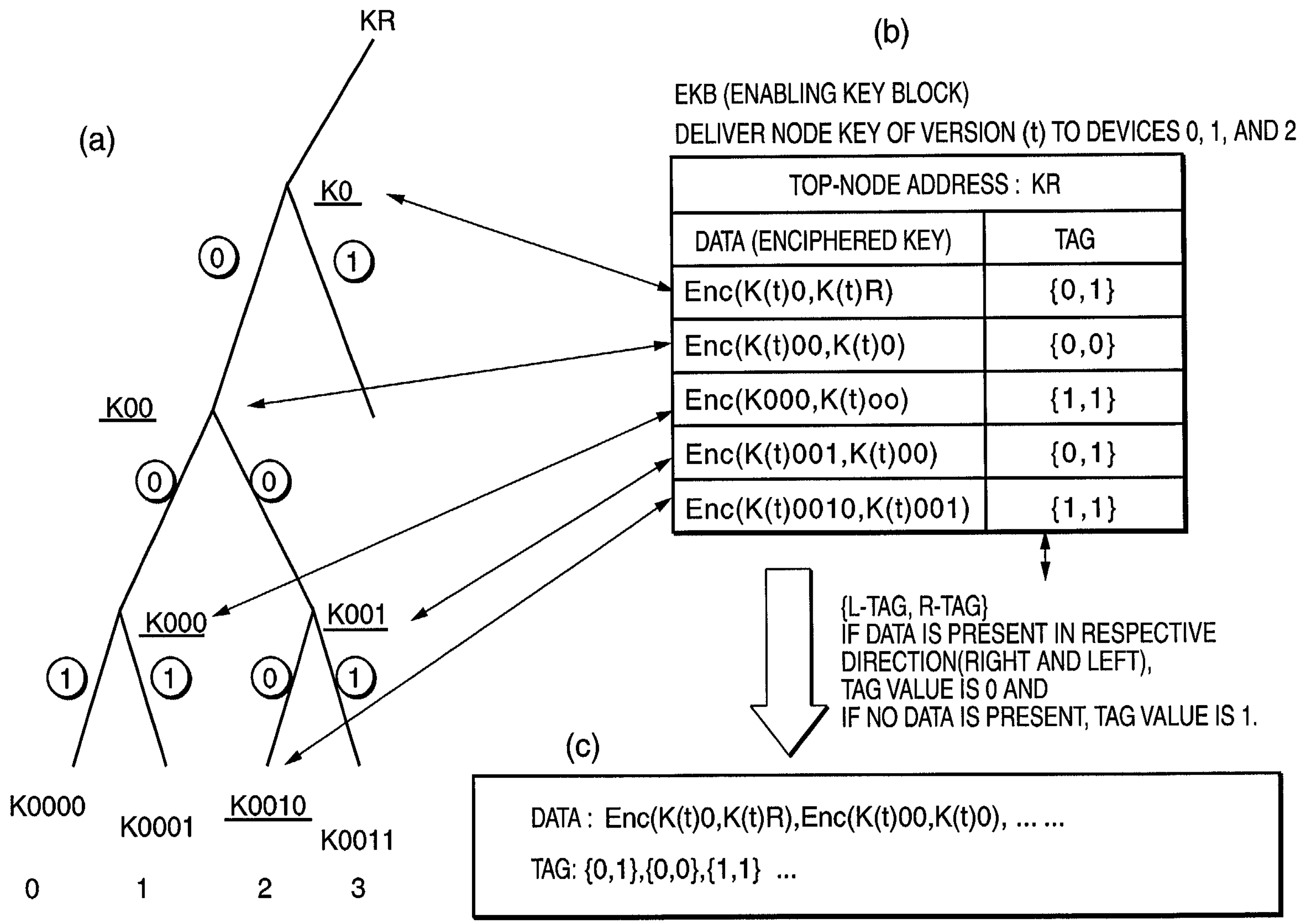


| VERSION : t | |
|---|---|
| INDEX | ENCIPHERING KEY |
| 000 | Enc(K000,K(t)00) |
| 001 | Enc(K(t)001,K(t)00) |
| 0010 | Enc(K0010,K(t)001) |
| Enc(K(t)00,K(t)con) | |

FIG. 6


601
VERSION
DEPTH
602
603
DATA POINTER
TAG POINTER
604
605
SIGNATURE POINTER
RESERVED
606
DATA BLOCK (E(KO, Kroot), ...    )
607
TAG BLOCK ({0, 0},{1, 1}, ...    )
608
SIGNATURE

# FIG. 7



EKB (ENABLING KEY BLOCK)
DELIVER NODE KEY OF VERSION (t) TO DEVICES 0, 1, AND 2

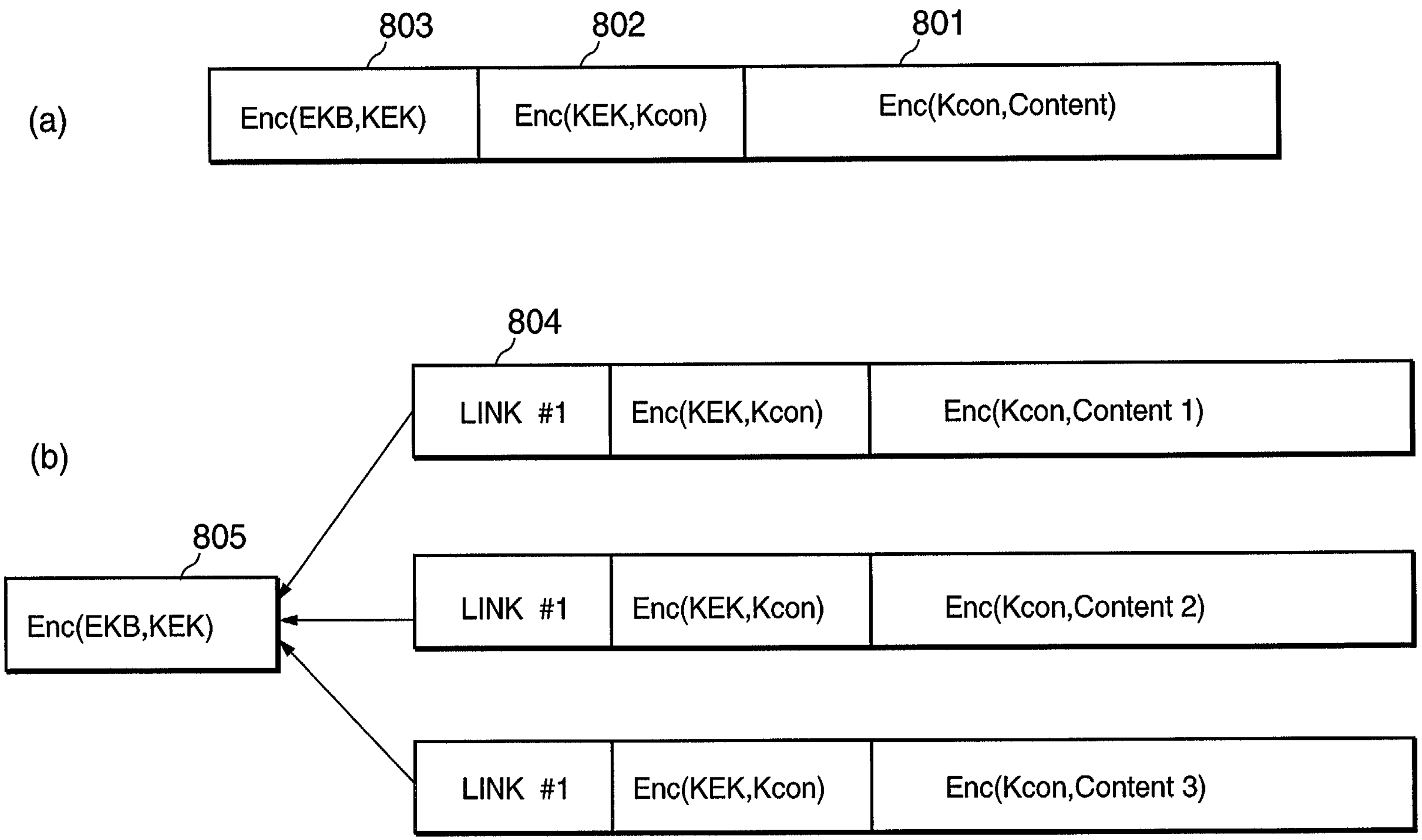
| TOP-NODE ADDRESS : KR | |
|---|---|
| DATA (ENCIPHERED KEY) | TAG |
| Enc(K(t)0,K(t)R) | {0,1} |
| Enc(K(t)00,K(t)0) | {0,0} |
| Enc(K000,K(t)oo) | {1,1} |
| Enc(K(t)001,K(t)00) | {0,1} |
| Enc(K(t)0010,K(t)001) | {1,1} |

{L-TAG, R-TAG}
IF DATA IS PRESENT IN RESPECTIVE DIRECTION(RIGHT AND LEFT), TAG VALUE IS 0 AND
IF NO DATA IS PRESENT, TAG VALUE IS 1.

DATA : Enc(K(t)0,K(t)R),Enc(K(t)00,K(t)0), ... ...

TAG: {0,1},{0,0},{1,1} ...

# FIG. 8


(a)
803
802
801
Enc(EKB,KEK)
Enc(KEK,Kcon)
Enc(Kcon,Content)
(b)
804
LINK #1
Enc(KEK,Kcon)
Enc(Kcon,Content 1)
LINK #1
Enc(KEK,Kcon)
Enc(Kcon,Content 2)
LINK #1
Enc(KEK,Kcon)
Enc(Kcon,Content 3)
805
Enc(EKB,KEK)

# FIG. 9

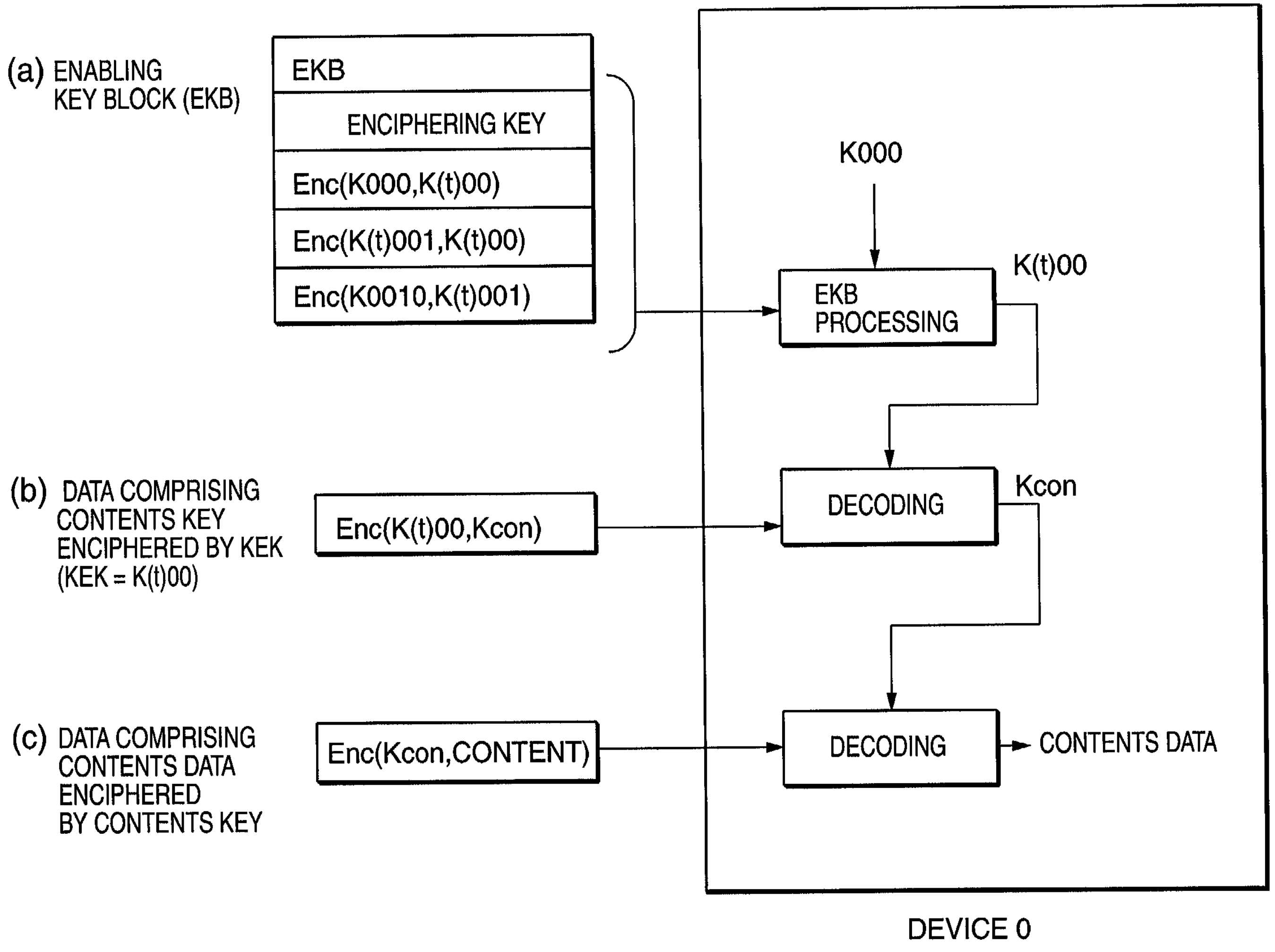
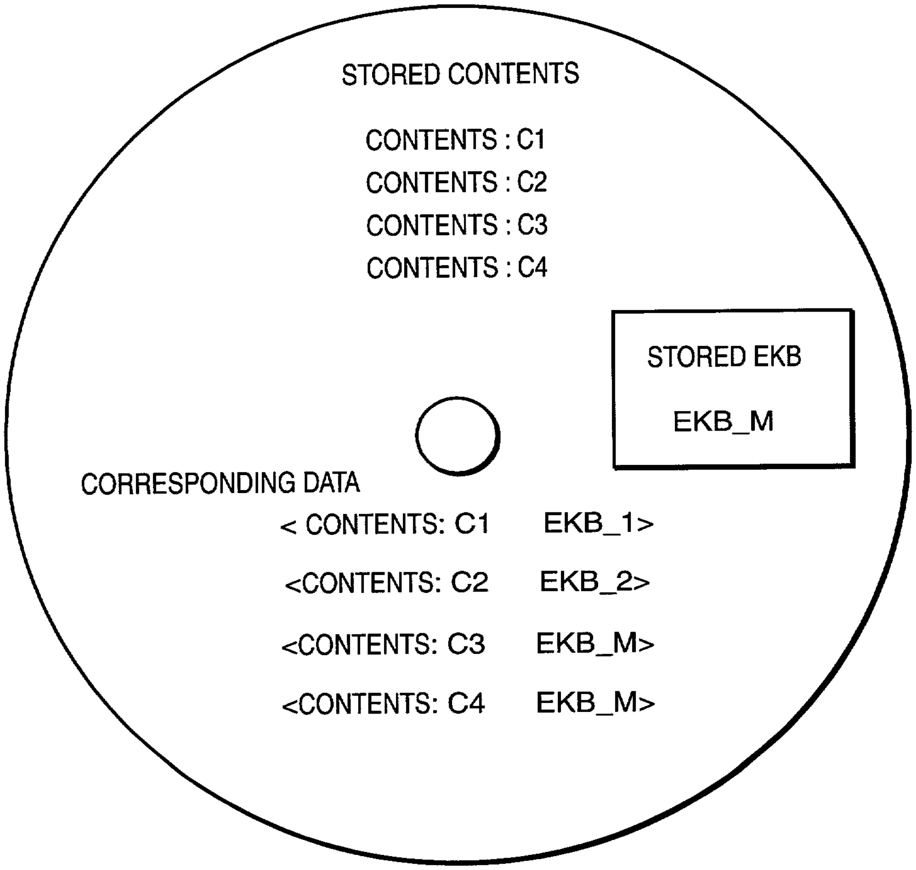
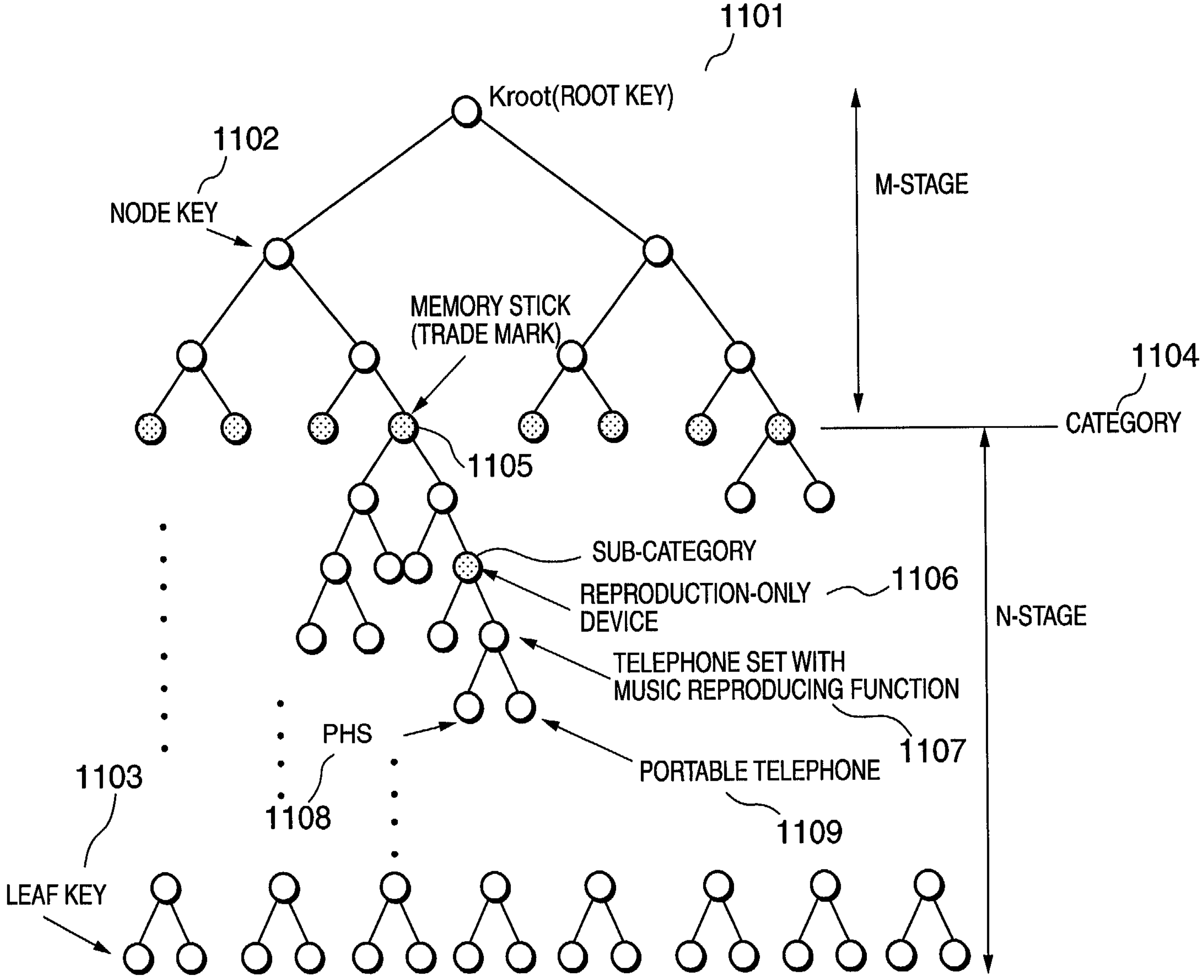

(a) ENABLING KEY BLOCK (EKB)
EKB
ENCIPHERING KEY
Enc(K000,K(t)00)
Enc(K(t)001,K(t)00)
Enc(K0010,K(t)001)
(b) DATA COMPRISING CONTENTS KEY ENCIPHERED BY KEK (KEK = K(t)00)
Enc(K(t)00,Kcon)
(c) DATA COMPRISING CONTENTS DATA ENCIPHERED BY CONTENTS KEY
Enc(Kcon,CONTENT)
K000
EKB PROCESSING
K(t)00
DECODING
Kcon
DECODING
CONTENTS DATA
DEVICE 0

FIG. 10

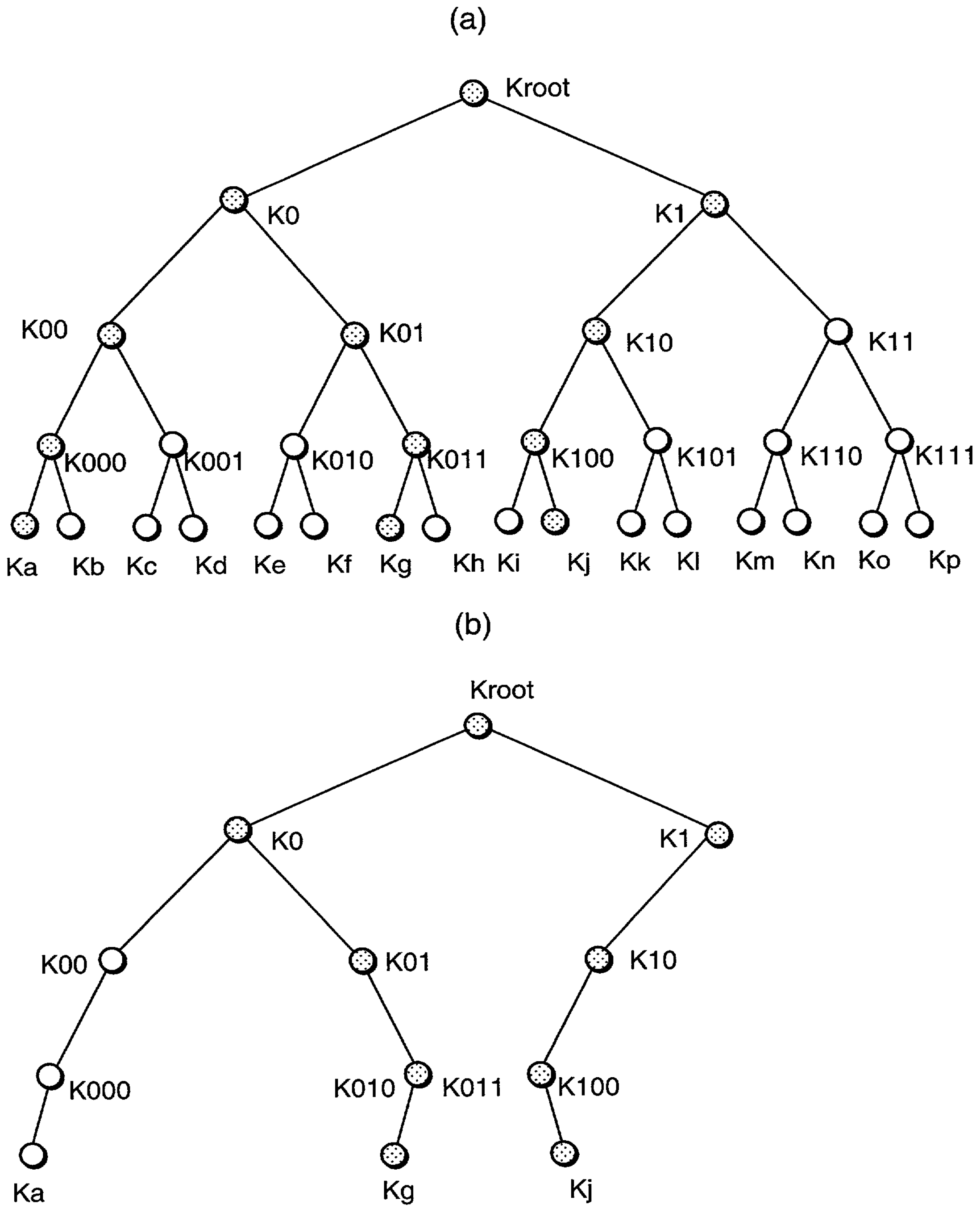


RECORDING MEDIUM

# FIG. 11


1101
Kroot(ROOT KEY)
1102
NODE KEY
M-STAGE
MEMORY STICK
(TRADE MARK)
1105
1104
CATEGORY
SUB-CATEGORY
REPRODUCTION-ONLY
DEVICE
1106
N-STAGE
TELEPHONE SET WITH
MUSIC REPRODUCING FUNCTION
1107
PHS
1108
PORTABLE TELEPHONE
1109
1103
LEAF KEY

# FIG. 12

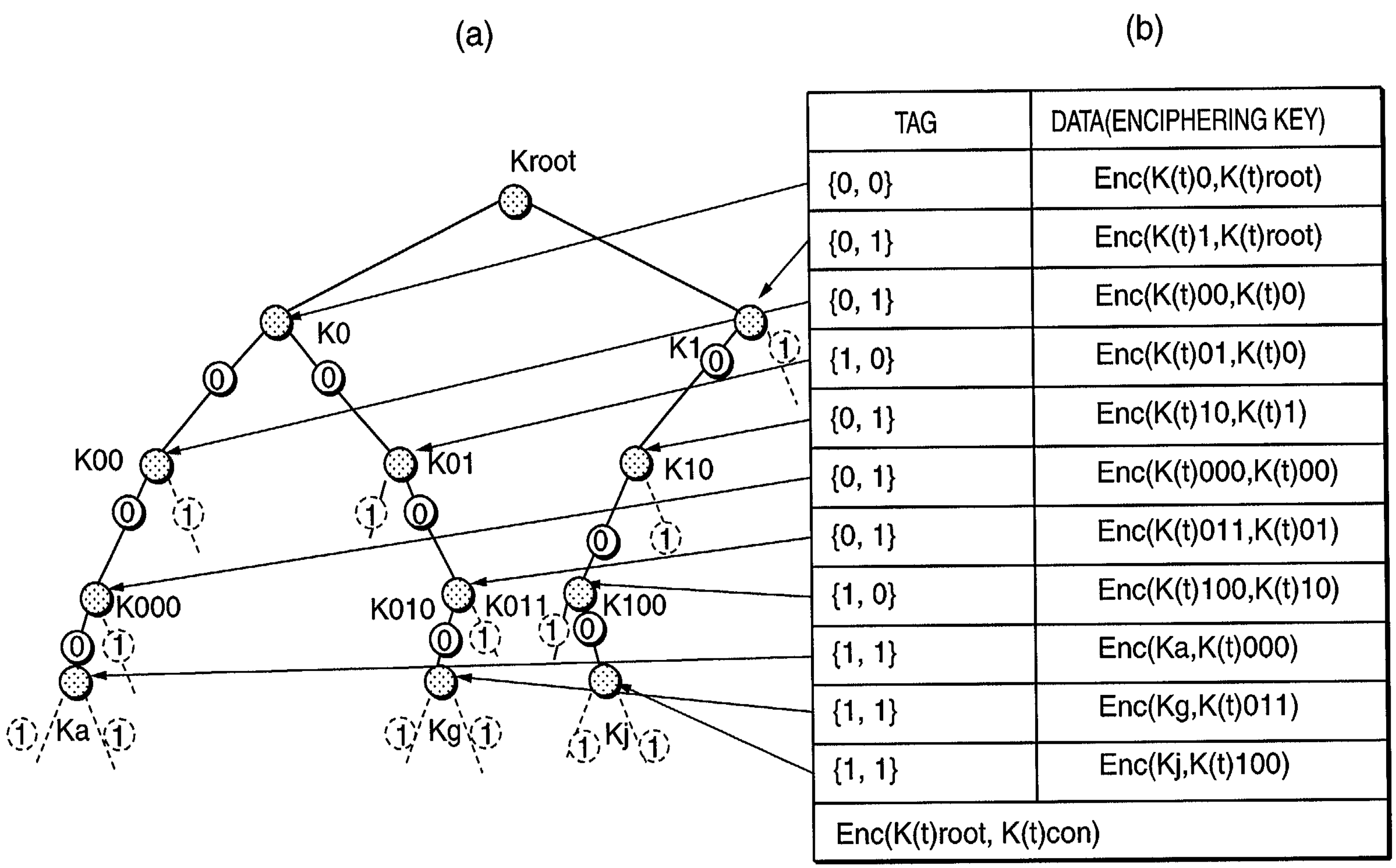


(a)

(b)

Kroot
K0
K1
K00
K01
K10
K11
K000
K001
K010
K011
K100
K101
K110
K111
Ka
Kb
Kc
Kd
Ke
Kf
Kg
Kh
Ki
Kj
Kk
Kl
Km
Kn
Ko
Kp
Kroot
K0
K1
K00
K01
K10
K000
K010
K011
K100
Ka
Kg
Kj

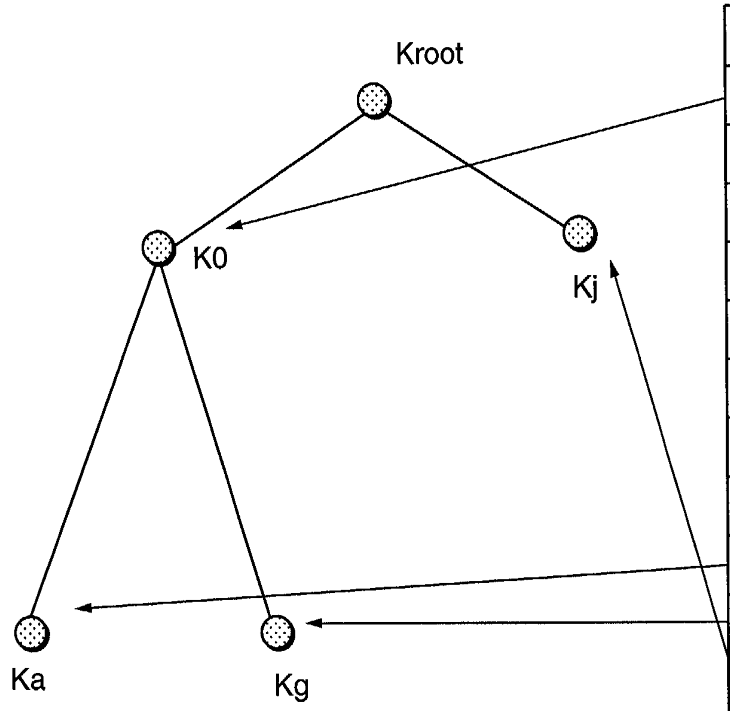
# FIG. 13



| TAG | DATA(ENCIPHERING KEY) |
|---|---|
| {0, 0} | Enc(K(t)0,K(t)root) |
| {0, 1} | Enc(K(t)1,K(t)root) |
| {0, 1} | Enc(K(t)00,K(t)0) |
| {1, 0} | Enc(K(t)01,K(t)0) |
| {0, 1} | Enc(K(t)10,K(t)1) |
| {0, 1} | Enc(K(t)000,K(t)00) |
| {0, 1} | Enc(K(t)011,K(t)01) |
| {1, 0} | Enc(K(t)100,K(t)10) |
| {1, 1} | Enc(Ka,K(t)000) |
| {1, 1} | Enc(Kg,K(t)011) |
| {1, 1} | Enc(Kj,K(t)100) |
| Enc(K(t)root, K(t)con) | |

# FIG. 14

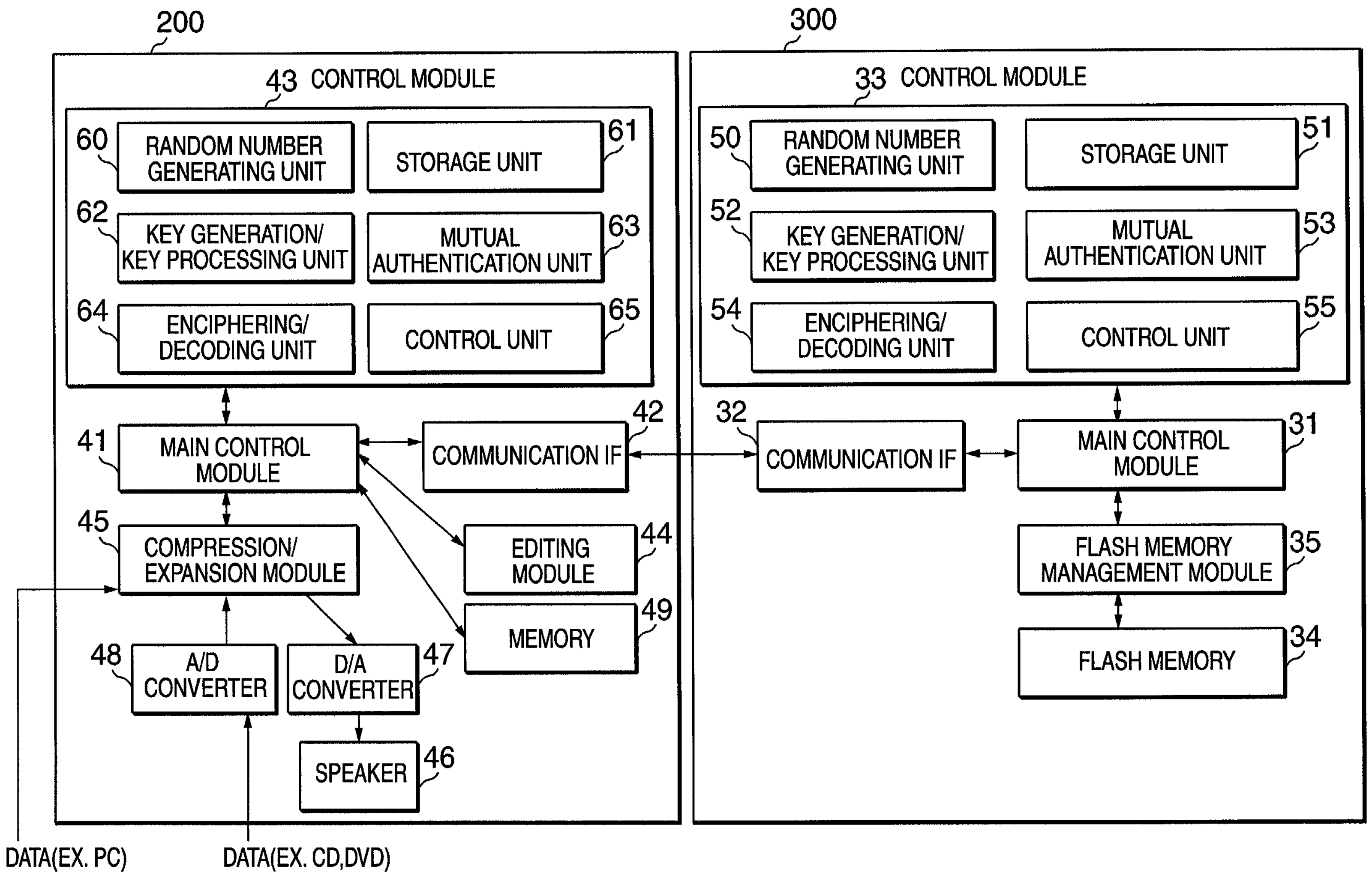
(a)



(b)

| TAG | DATA(ENCIPHERING KEY) |
|---|---|
| {0, 0, 1} | Enc(K(t)0,K(t)root) |
| {0, 1, 0} | — |
| {0, 1, 0} | — |
| {1, 0, 0} | — |
| {0, 1, 0} | — |
| {0, 1, 0} | — |
| {0, 1, 0} | — |
| {1, 0, 0} | — |
| {1, 1, 1} | Enc(Ka,K(t)0) |
| {1, 1, 1} | Enc(Kg,K(t)0) |
| {1, 1, 1} | Enc(Kj,K(t)root) |
| Enc(K(t)root, K(t)con) | |

# FIG. 15


200
43 CONTROL MODULE
60
RANDOM NUMBER GENERATING UNIT
STORAGE UNIT
61
62
KEY GENERATION/ KEY PROCESSING UNIT
MUTUAL AUTHENTICATION UNIT
63
64
ENCIPHERING/ DECODING UNIT
CONTROL UNIT
65
41
MAIN CONTROL MODULE
COMMUNICATION IF
42
45
COMPRESSION/ EXPANSION MODULE
EDITING MODULE
44
MEMORY
49
48
A/D CONVERTER
D/A CONVERTER
47
SPEAKER
46
DATA(EX. PC)
DATA(EX. CD,DVD)
300
33 CONTROL MODULE
50
RANDOM NUMBER GENERATING UNIT
STORAGE UNIT
51
52
KEY GENERATION/ KEY PROCESSING UNIT
MUTUAL AUTHENTICATION UNIT
53
54
ENCIPHERING/ DECODING UNIT
CONTROL UNIT
55
32
COMMUNICATION IF
MAIN CONTROL MODULE
31
FLASH MEMORY MANAGEMENT MODULE
35
FLASH MEMORY
34

# FIG. 16

DATA STORED IN A STORAGE UNIT OF A MEMORY DEVICE

| | |
|---|---|
| AUTHENTICATION KEY DATA | IK0<br>IK1<br>IK2<br>IK3<br>:<br>:<br>IK30<br>IK31 |
| DEVICE IDENTIFICATION DATA | ID0 |
| STORAGE KEY DATA | Kstm |

FIG. 17

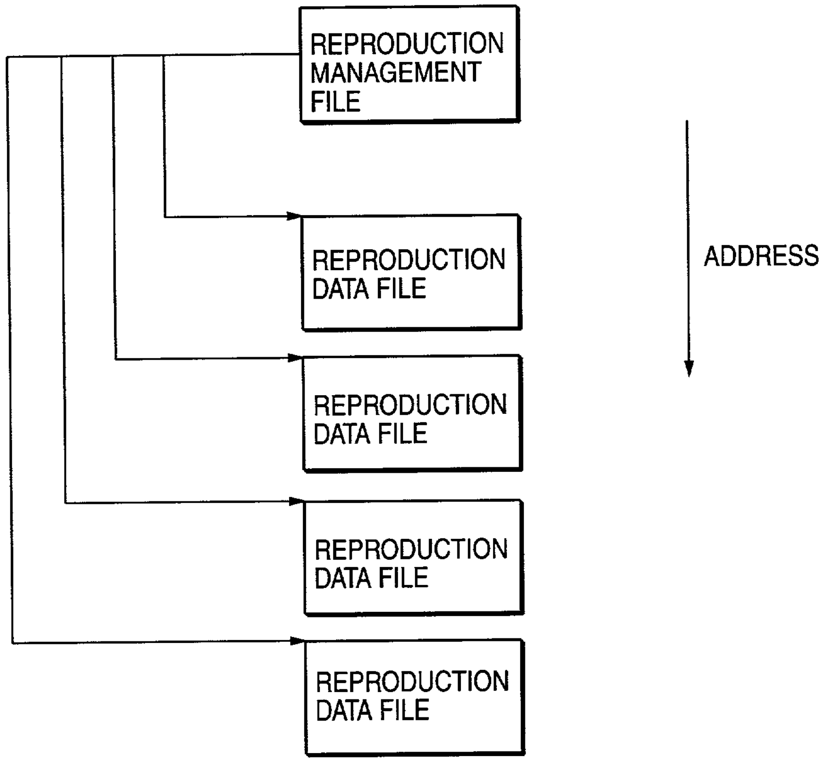

REPRODUCTION MANAGEMENT FILE
REPRODUCTION DATA FILE
REPRODUCTION DATA FILE
REPRODUCTION DATA FILE
REPRODUCTION DATA FILE
ADDRESS

FIG. 18

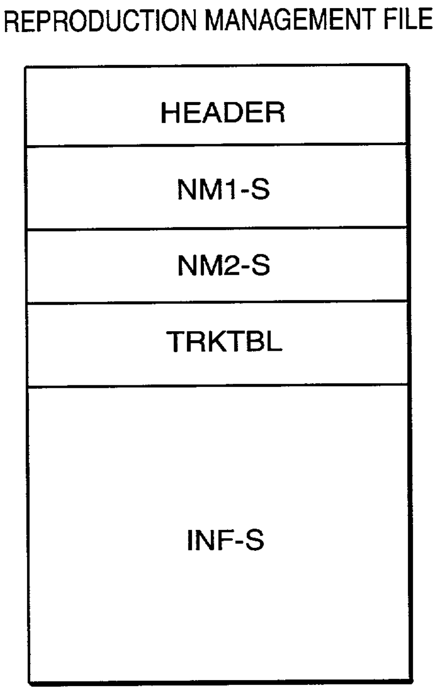
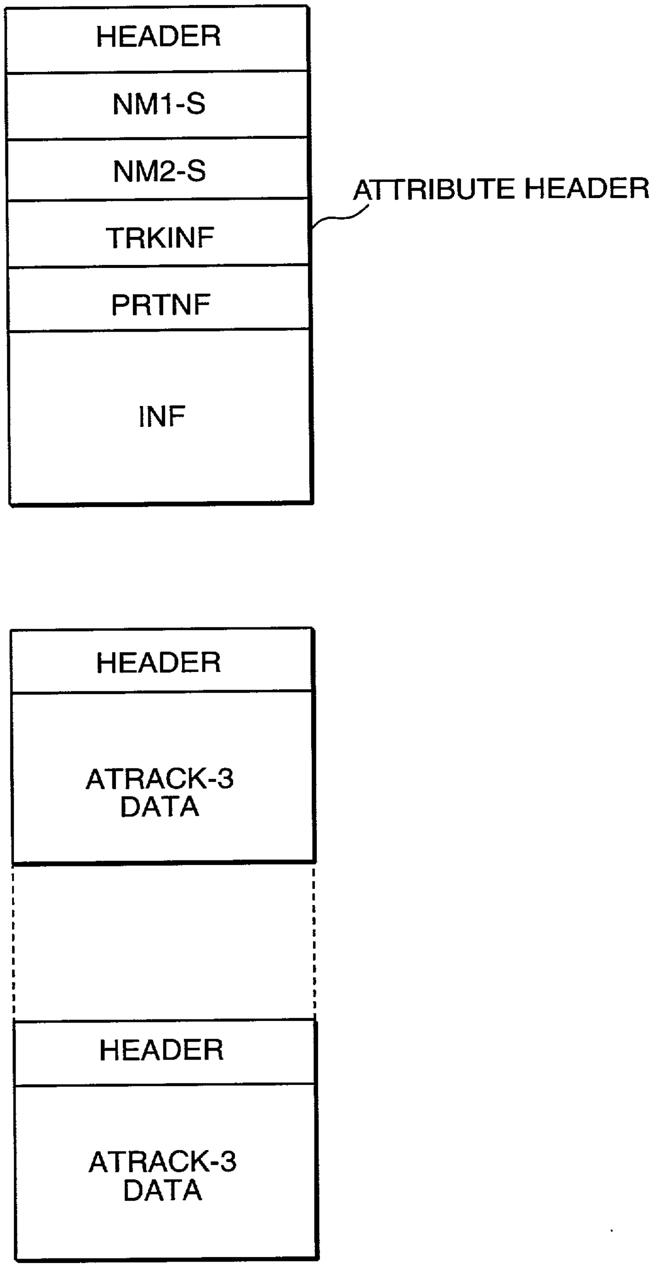

REPRODUCTION MANAGEMENT FILE
HEADER
NM1-S
NM2-S
TRKTBL
INF-S

FIG. 19


HEADER
NM1-S
NM2-S
TRKINF
PRTNF
INF
ATTRIBUTE HEADER
HEADER
ATRACK-3 DATA
HEADER
ATRACK-3 DATA

# FIG. 20

REPRODUCTION MANAGEMENT FILE

A

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x0000 | BLKID-TLO | | | | RESERVED | | MCODE | | REVISION | | | | RESERVED | | | |
| 0x0010 | SN1C+L | | SN2C+L | | SINFSIZE | | T-TRK | | VerNo. | | RESERVED | | | | | |

B

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x0020 | NM1-S(256) | | | | | | | | | | | | | | | |
| 0x0120 | NM2-S(512) | | | | | | | | | | | | | | | |
| 0x0310 | | | | | | | | | | | | | | | | |
| 0x0320 | RESERVED(4) | | | | EKB VERSION | | | | E(Kstm,Kcon) | | | | | | | |
| 0x0330 | E(KEKn,Kcon) | | | | | | | | c_MAC[0] | | | | | | | |
| 0x0340 | RESERVED(8) | | | | | | | | RESERVED(3) | | | MGR | S-YMDhms | | | |
| 0x0350 | TRK-001 | | TRK-002 | | TRK-003 | | TRK-004 | | TRK-005 | | TRK-006 | | TRK-007 | | TRK-008 | |
| 0x0360 | TRK-009 | | TRK-010 | | TRK-011 | | TRK-012 | | TRK-013 | | TRK-014 | | TRK-015 | | TRK-016 | |
| | | | | | | | | | | | | | | | | |
| 0x0660 | TRK-393 | | TRK-394 | | TRK-395 | | TRK-396 | | TRK-397 | | TRK-398 | | TRK-399 | | TRK-400 | |
| 0x0670 | INF-S(14720) | | | | | | | | | | | | | | | |
| 0x3FFF | BLKID-TLO | | | | RESERVED | | MCODE | | REVISION | | | | RESERVED | | | |

C

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| INF | 0X00 | ID | 0X00 | SIZE | | MCODE | | C+L | | RESERVED | | DATA VARIABLE LENGTH | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

# FIG. 21

ATRACK-3 DATA FILE

<table>
<tr><th></th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th><th>F</th></tr>
<tr><td>0x0000</td><td colspan="4">BLKID-HDO</td><td colspan="2">RESERVED</td><td colspan="2">MCODE</td><td colspan="4">RESERVED</td><td colspan="4">BLOCK SERIAL</td></tr>
<tr><td>0x0010</td><td colspan="2">N1C+L</td><td colspan="2">N2C+L</td><td colspan="2">INFSIZE</td><td colspan="2">T-PRT</td><td colspan="4">T-SU</td><td colspan="2">INX</td><td colspan="2">XT</td></tr>
<tr><td>0x0020</td><td colspan="16">NM1-S(256)</td></tr>
<tr><td>0x0120</td><td colspan="16">NM2-S(512)</td></tr>
<tr><td>0x0310</td><td colspan="16"></td></tr>
<tr><td>0x0320</td><td colspan="3">RESERVED(3)</td><td colspan="1">EKI</td><td colspan="4">EKB VERSION</td><td colspan="8">E(Kstm, Kcon)</td></tr>
<tr><td>0x0330</td><td colspan="8">E(KEKn. Kcon)</td><td colspan="8">C_MAC[n]</td></tr>
<tr><td>0x0340</td><td colspan="8">RESERVED(8)</td><td colspan="4">INF_seq#</td><td>A</td><td>LT</td><td colspan="2">FNo</td></tr>
<tr><td>0x0350</td><td colspan="8">MG(D)SERIAL-nnn(Upper)</td><td colspan="8">MG(D)SERIAL-nnn(LOWER)</td></tr>
<tr><td>0x0360</td><td colspan="4">CONNUM</td><td colspan="4">YMDhms-S</td><td colspan="4">YMDhms-E</td><td>XCC</td><td>CT</td><td>CC</td><td>CN</td></tr>
<tr><td>0x0370</td><td colspan="4">PRTSIZE</td><td colspan="8">PRTKEY</td><td colspan="4">RESERVED(8)</td></tr>
<tr><td>0x0380</td><td colspan="4"></td><td colspan="4">CONNUMO</td><td colspan="4">PRTSIZE(0x0388)</td><td colspan="4">PRTKEY</td></tr>
<tr><td>0x0390</td><td colspan="4"></td><td colspan="8">RESERVED(8)</td><td colspan="4">CONNUMO</td></tr>
<tr><td></td><td colspan="16">INF(0x0400)</td></tr>
<tr><td>0x3FFF</td><td colspan="4">BLKID-HDD</td><td colspan="2">RESERVED</td><td colspan="2">MCODE</td><td colspan="4">RESERVED</td><td colspan="4">BLOCK SERIAL</td></tr>
<tr><td>0x4000</td><td colspan="4">BLKID-A3D</td><td colspan="2">RESERVED</td><td colspan="2">MCODE</td><td colspan="4">CONNUMO</td><td colspan="4">BLOCK SERIAL</td></tr>
<tr><td>0x4010</td><td colspan="8">BLOCKSEED</td><td colspan="8">INTIALIZATION VECTOR</td></tr>
<tr><td>0x4020</td><td colspan="16">SU-000(NByte=384Byte)</td></tr>
<tr><td>0x41A0</td><td colspan="16">SU-001(NByte)</td></tr>
<tr><td>0x4320</td><td colspan="16">SU-002(NByte)</td></tr>
<tr><td>0x04A0</td><td colspan="16">SU-041(NByte)</td></tr>
<tr><td>0x7DA0</td><td colspan="16">RESERVED(NByte=208Byte)</td></tr>
<tr><td>0x7F20</td><td colspan="8">BLK SEED</td><td colspan="8"></td></tr>
<tr><td>0x7FF0</td><td colspan="4">BLKID-A3D</td><td colspan="2">RESERVED</td><td colspan="2">MCODE</td><td colspan="4">CONNUMO</td><td colspan="4">BLOCK SERIAL</td></tr>
</table>

FIG. 22

0 1 2 3 4 5 6 7 8 9 A B C D E F

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0x0000 | BLKID-HDO | | RESERVED | MCODE | RESERVED | BLOCK SERIAL | |
| 0x0010 | N1C+L | N2C+L | INFSIZE | T-PRT | T-SU | INX | XT |
| 0x0020 | NM1-S(256) | | | | | | |
| 0x0120 | NM2-S(512) | | | | | | |
| 0x0310 | | | | | | | |

# FIG. 23

<table>
<tr><td>0x0320</td><td>RESERVED(3)</td><td>EKI</td><td>EKB VERSION</td><td colspan="7">E(Kstm, Kcon)</td></tr>
<tr><td>0x0330</td><td colspan="3">E(KEKn, Kcon)</td><td colspan="7">C_MAC[n]</td></tr>
<tr><td>0x0340</td><td colspan="3">RESERVED(8)</td><td>INF_seq#</td><td colspan="2">A</td><td colspan="2">LT</td><td colspan="2">FNo</td></tr>
<tr><td>0x0350</td><td colspan="3">MG(D)SERIAL-nnn(UPPER)</td><td colspan="7">MG(D)SERIAL-nnn(LOWER)</td></tr>
<tr><td>0x0360</td><td colspan="2">CONNUM</td><td>YMDhms-S</td><td>YMDhms-E</td><td colspan="2">XCC</td><td colspan="2">CT</td><td>CC</td><td>CN</td></tr>
</table>

# FIG. 24

| | | |
|---|---|---|
| Bit7 : ATRAC3 Mode | 0 : Dual | 1 : Joint |

Bits 6, 5, 4: N OF 3-Bit CORRESPONDS TO MODE VALUE

| N | MODE | TIME | TRANSFER RATE | SU (SOUND UNIT) | Byte |
|---|---|---|---|---|---|
| 7 | HQ | 47min | 176kbps | 31SU | 512 |
| 6 | | 58min | 146kbps | 38SU | 424 |
| 5 | EX | 64min | 132kbps | 42SU | 384 |
| 4 | SP | 81min | 105kbps | 53SU | 304 |
| 3 | | 90min | 94kbps | 59SU | 272 |
| 2 | LP | 128min | 66kbps | 84SU | 192 |
| 1 | MONO | 181min | 47kbps | 119SU | 136 |
| 0 | MONO | 258min | 33kbps | 169SU | 96 |

| | | |
|---|---|---|
| Bit3 : RESERVED | | |
| Bit2 : DATA DISTINCTION | 0 : AUDIO | 1 : OTHERS |
| Bit1 : REPRODUCED SKIP | 0 : NORMAL REPRODUCTION | 1 : SKIP |
| Bit0 : EMPHASIS | 0 : OFF | 1 : ON(50/15 $\mu$ SECCOND) |

# FIG. 25

| | | | |
|---|---|---|---|
| | Bit7 : COPY APPROVAL | 0 : COPY INHIBITED | 1 : COPY APPROVED |
| | Bit6 : GENERATION (VERSION) | 0 : ORIGINAL | 1 : BEYOND THE FIRST GENERATION |

HCMS Bit5-4 : CONTROL IN RELATION TO HIGH-SPEED DIGITAL COPYING OPERATION

00 : COPY INHIBITED    01 : COPY FOR THE FIRST GENERATION 10 : COPY APPROVED

CHILD WHO IMPLEMENTED COPYING OF THE FIRST GENERATION IS INHIBITED FROM EXECUTING FURTHER COPYING OPERASTION

Bit3-2 : MAGIC GATE AUTHENTICATION LEVEL

| | |
|---|---|
| 00: LEVEL10(Non-MG) | 01 : LEVEL1 |
| 02: LEVEL12 | 11 : RESERVED |
| 02: LEVEL10 | |

THOSE LEVELS OTHER THAN 10 CAN NOT BE DIVIDED NOR COMBINED

Bit1, 0 : RESERVED

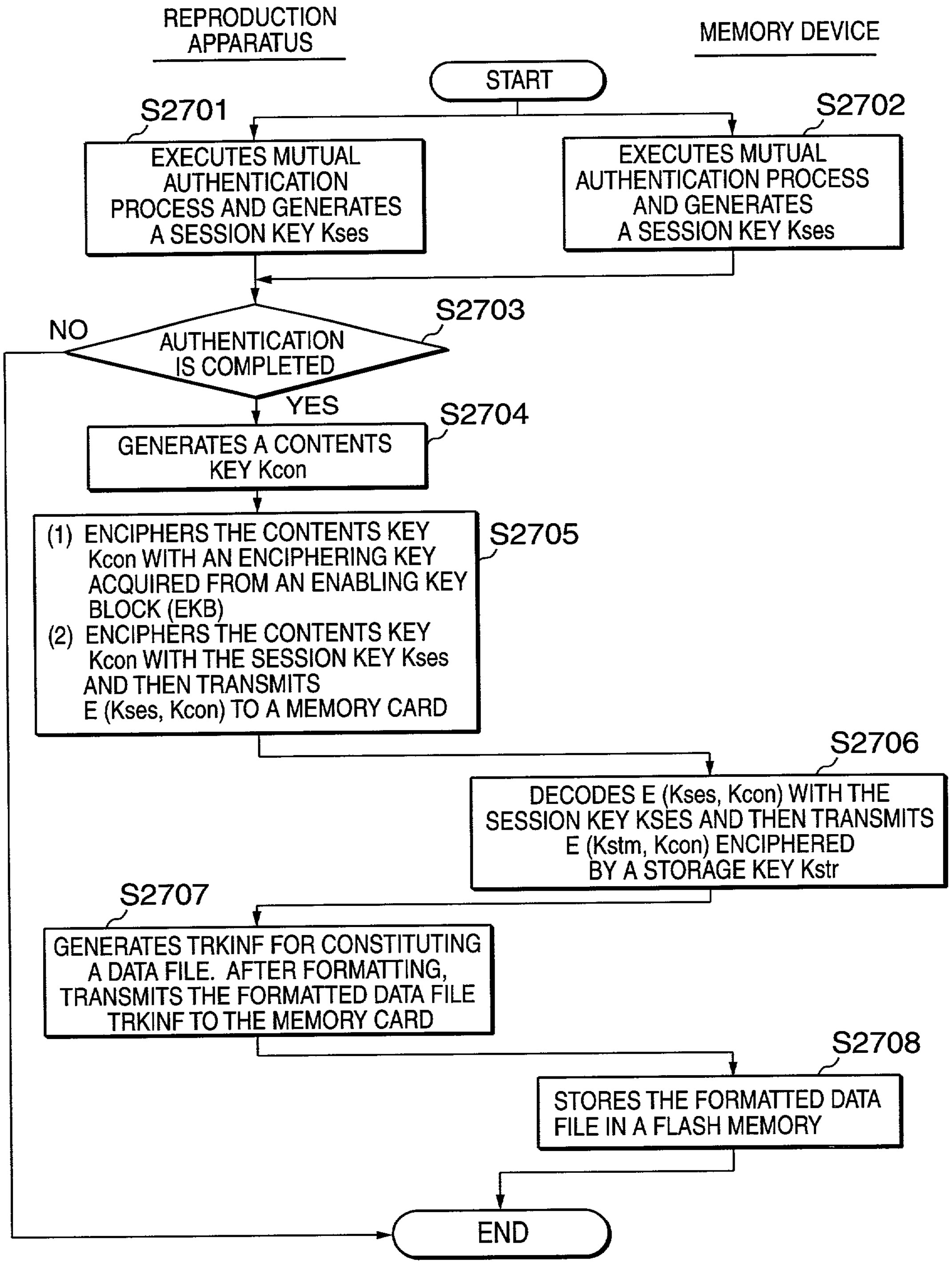
# FIG. 26

|  |  |  |  |  |
|---|---|---|---|---|
| 0x0370 | PRTSIZE | PRTKEY |  | RESERVED (8) |
| 0x0380 |  | CONNUMO | PRTSIZE(0x0388) | PRTKEY |
| 0x0390 |  | RESERVED (8) |  | CONNUMO |

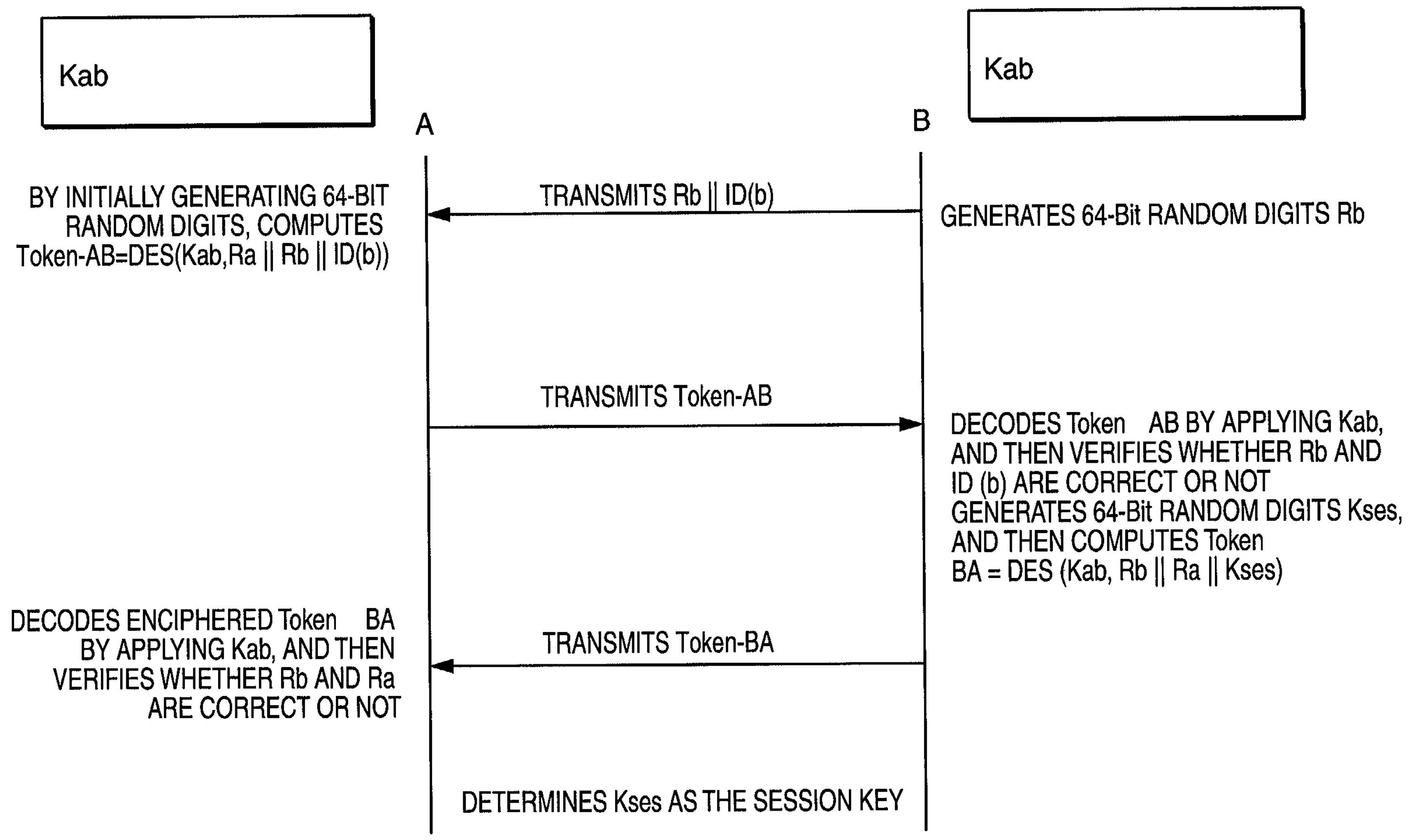
# FIG. 27

|  |  |  |  |  |  |
|---|---|---|---|---|---|
| 0x4000 | BLKID-A3D | RESERVED | MCODE | CONNUMO | BLOCK SERIAL |
| 0x4010 | BLOCKSEED |  |  | INTIALIZATION VECTOR |  |
| 0x4020 | SU-000(NByte=384Byte) |  |  |  |  |

FIG. 28

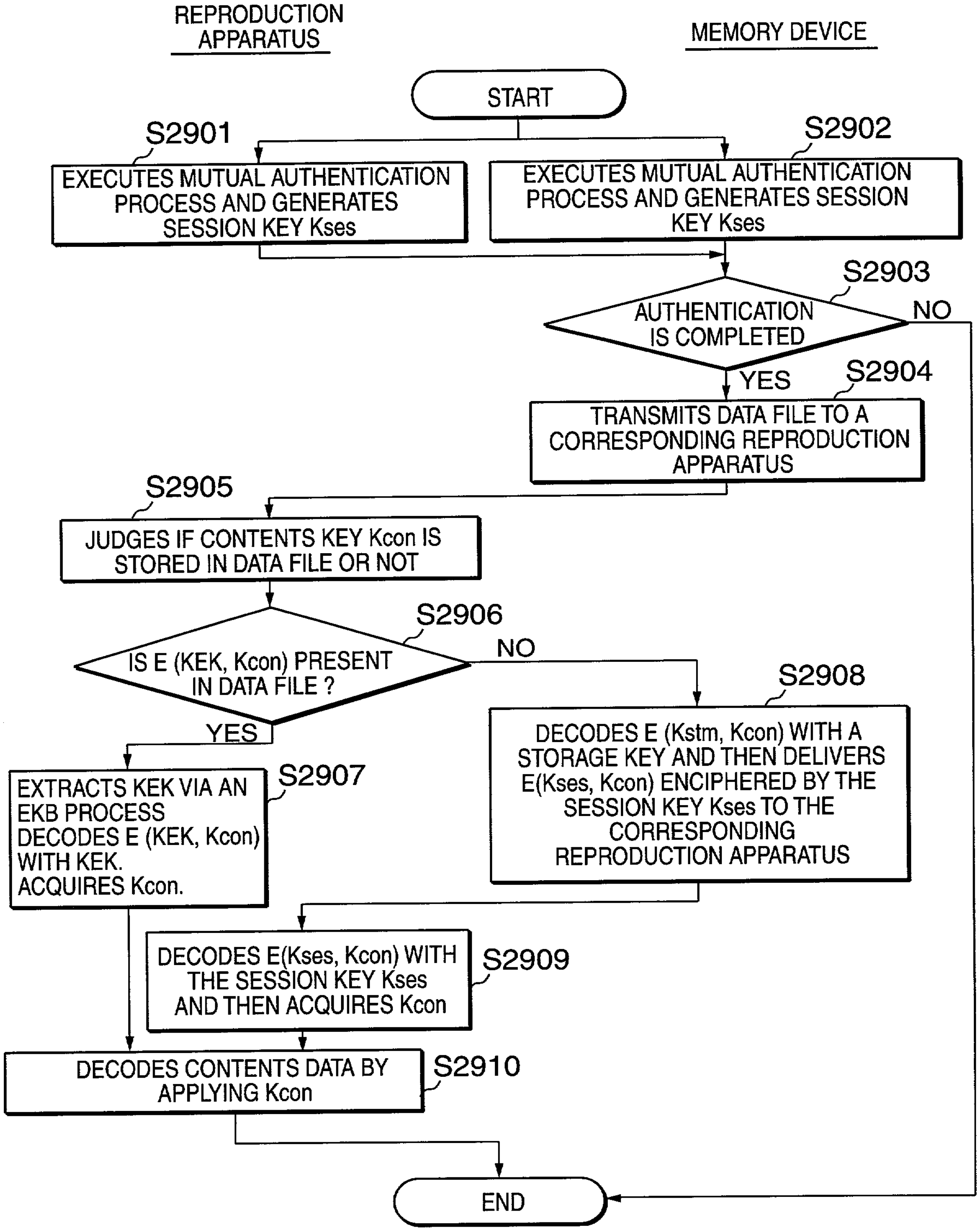

REPRODUCTION APPARATUS
MEMORY DEVICE
START
S2701
EXECUTES MUTUAL AUTHENTICATION PROCESS AND GENERATES A SESSION KEY Kses
S2702
EXECUTES MUTUAL AUTHENTICATION PROCESS AND GENERATES A SESSION KEY Kses
S2703
AUTHENTICATION IS COMPLETED
NO
YES
S2704
GENERATES A CONTENTS KEY Kcon
S2705
(1) ENCIPHERS THE CONTENTS KEY Kcon WITH AN ENCIPHERING KEY ACQUIRED FROM AN ENABLING KEY BLOCK (EKB)
(2) ENCIPHERS THE CONTENTS KEY Kcon WITH THE SESSION KEY Kses AND THEN TRANSMITS E (Kses, Kcon) TO A MEMORY CARD
S2706
DECODES E (Kses, Kcon) WITH THE SESSION KEY KSES AND THEN TRANSMITS E (Kstm, Kcon) ENCIPHERED BY A STORAGE KEY Kstr
S2707
GENERATES TRKINF FOR CONSTITUTING A DATA FILE. AFTER FORMATTING, TRANSMITS THE FORMATTED DATA FILE TRKINF TO THE MEMORY CARD
S2708
STORES THE FORMATTED DATA FILE IN A FLASH MEMORY
END

# FIG. 29


Kab
Kab
A
B
BY INITIALLY GENERATING 64-BIT RANDOM DIGITS, COMPUTES Token-AB=DES(Kab,Ra || Rb || ID(b))
TRANSMITS Rb || ID(b)
GENERATES 64-Bit RANDOM DIGITS Rb
TRANSMITS Token-AB
DECODES Token AB BY APPLYING Kab, AND THEN VERIFIES WHETHER Rb AND ID (b) ARE CORRECT OR NOT GENERATES 64-Bit RANDOM DIGITS Kses, AND THEN COMPUTES Token BA = DES (Kab, Rb || Ra || Kses)
DECODES ENCIPHERED Token BA BY APPLYING Kab, AND THEN VERIFIES WHETHER Rb AND Ra ARE CORRECT OR NOT
TRANSMITS Token-BA
DETERMINES Kses AS THE SESSION KEY

MUTUAL AUTHENTICATION FORMAT AND KEY-COMMUNIZING FORMAT VIA UTILIZATION OF THE ISO/IEC9798-2 STANDARD SYMMETRICAL KEY ENCIPHERING ART

FIG. 30


REPRODUCTION APPARATUS
MEMORY DEVICE
START
S2901
EXECUTES MUTUAL AUTHENTICATION PROCESS AND GENERATES SESSION KEY Kses
S2902
EXECUTES MUTUAL AUTHENTICATION PROCESS AND GENERATES SESSION KEY Kses
S2903
AUTHENTICATION IS COMPLETED
NO
YES
S2904
TRANSMITS DATA FILE TO A CORRESPONDING REPRODUCTION APPARATUS
S2905
JUDGES IF CONTENTS KEY Kcon IS STORED IN DATA FILE OR NOT
S2906
IS E (KEK, Kcon) PRESENT IN DATA FILE ?
NO
YES
S2907
EXTRACTS KEK VIA AN EKB PROCESS DECODES E (KEK, Kcon) WITH KEK. ACQUIRES Kcon.
S2908
DECODES E (Kstm, Kcon) WITH A STORAGE KEY AND THEN DELIVERS E(Kses, Kcon) ENCIPHERED BY THE SESSION KEY Kses TO THE CORRESPONDING REPRODUCTION APPARATUS
S2909
DECODES E(Kses, Kcon) WITH THE SESSION KEY Kses AND THEN ACQUIRES Kcon
S2910
DECODES CONTENTS DATA BY APPLYING Kcon
END

# FIG. 31

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x0000 | BLKID-EKB | | | | RESERVED | | MCODE | | RESERVED(3) | | | LKF | LINK COUNT | | | |
| 0x0010 | RESERVED(8) | | | | | | | | RESERVED(8) | | | | | | | |
| 0x0020 | VERSION | | | | EA | RESERVED | | | KEK1 | | | | | | | |
| 0x0030 | KEK2 | | | | | | | | E(VERSION) | | | | | | | |
| 0x0040 | SIZE OF TAG PART | | | | SIZE OF KEY PART | | | | SIZE OF SIGN PART | | | | | | | |
| 0x0050 | TAG PART ({X,O,O}, {X,1,1}. . . . . . . . . .)<br>FILL TO 64Bit ALIGNMENT | | | | | | | | | | | | | | | |
| | KEY PART | | | | | | | | | | | | | | | |
| | SIGNATURE | | | | | | | | | | | | | | | |

FIG. 32

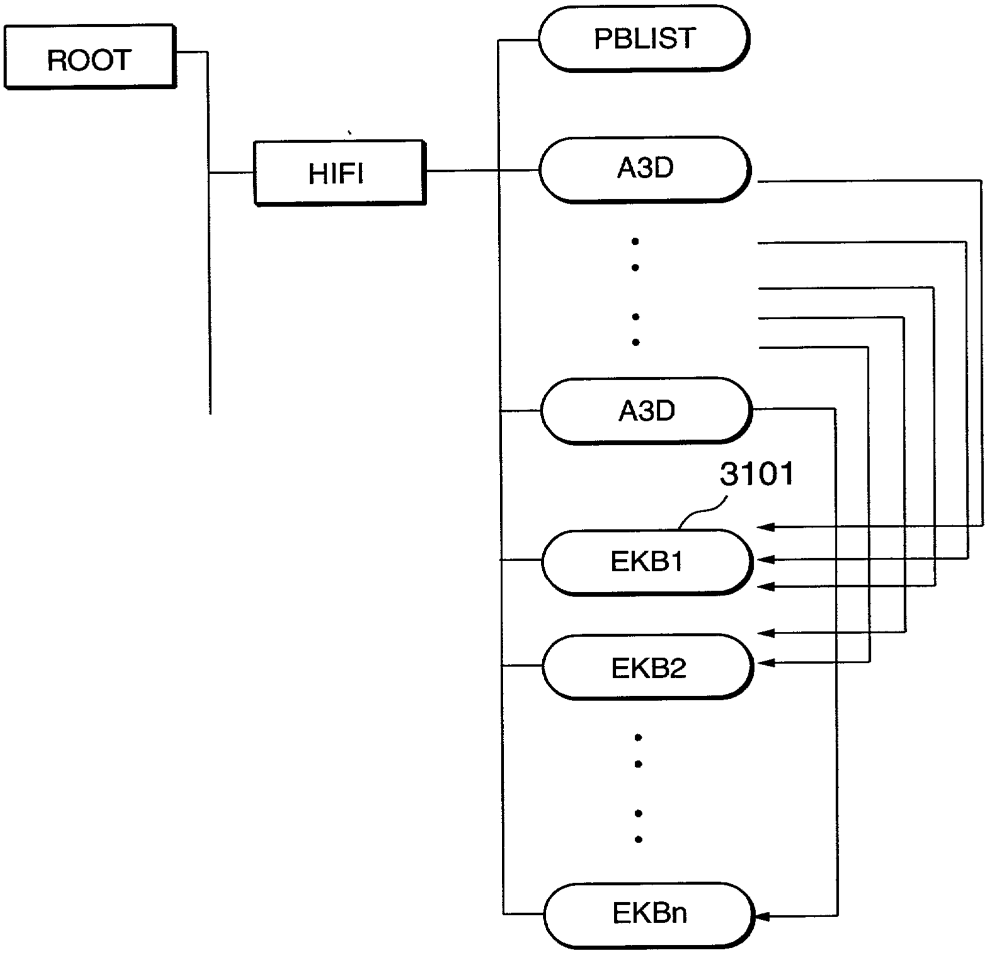
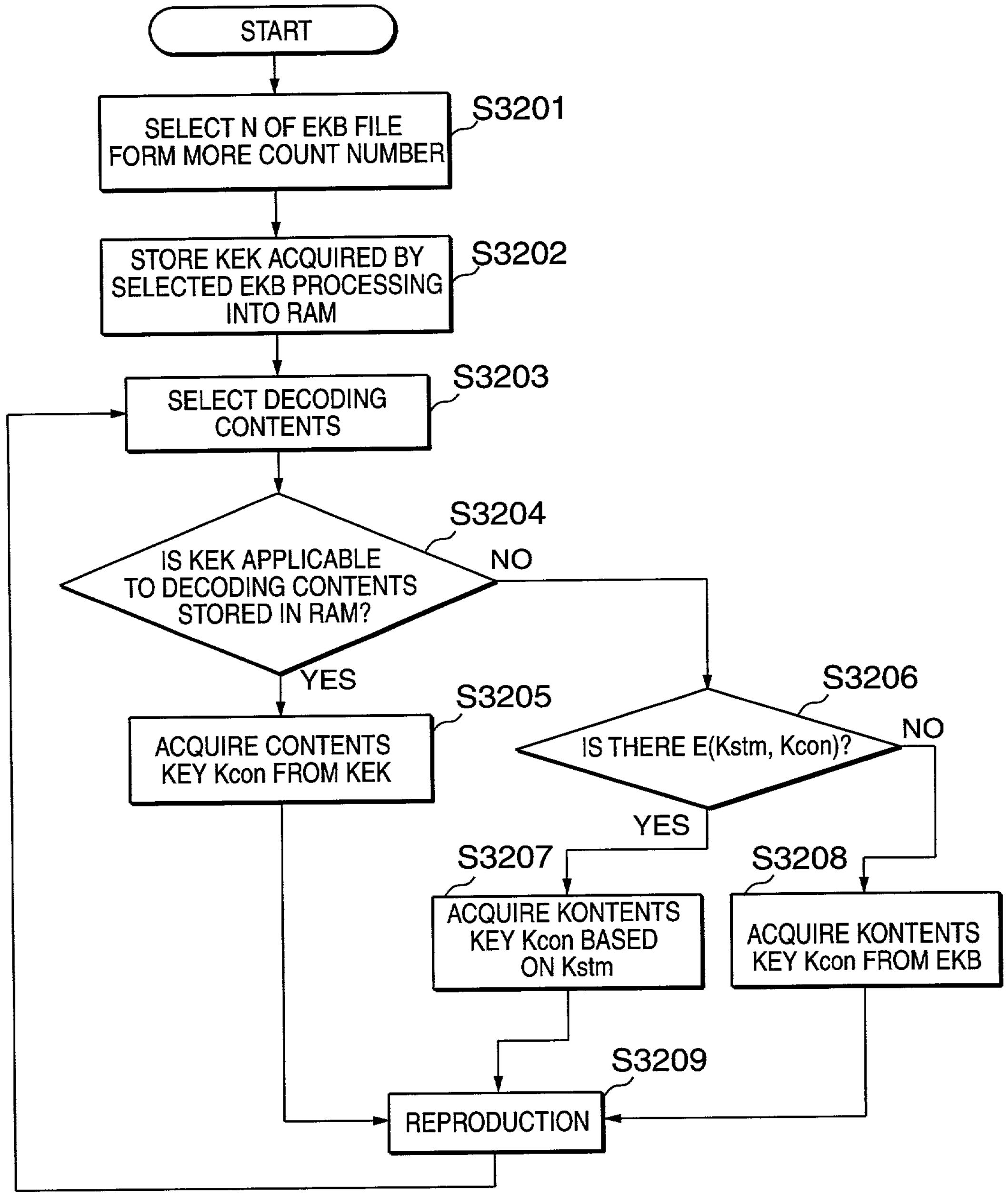
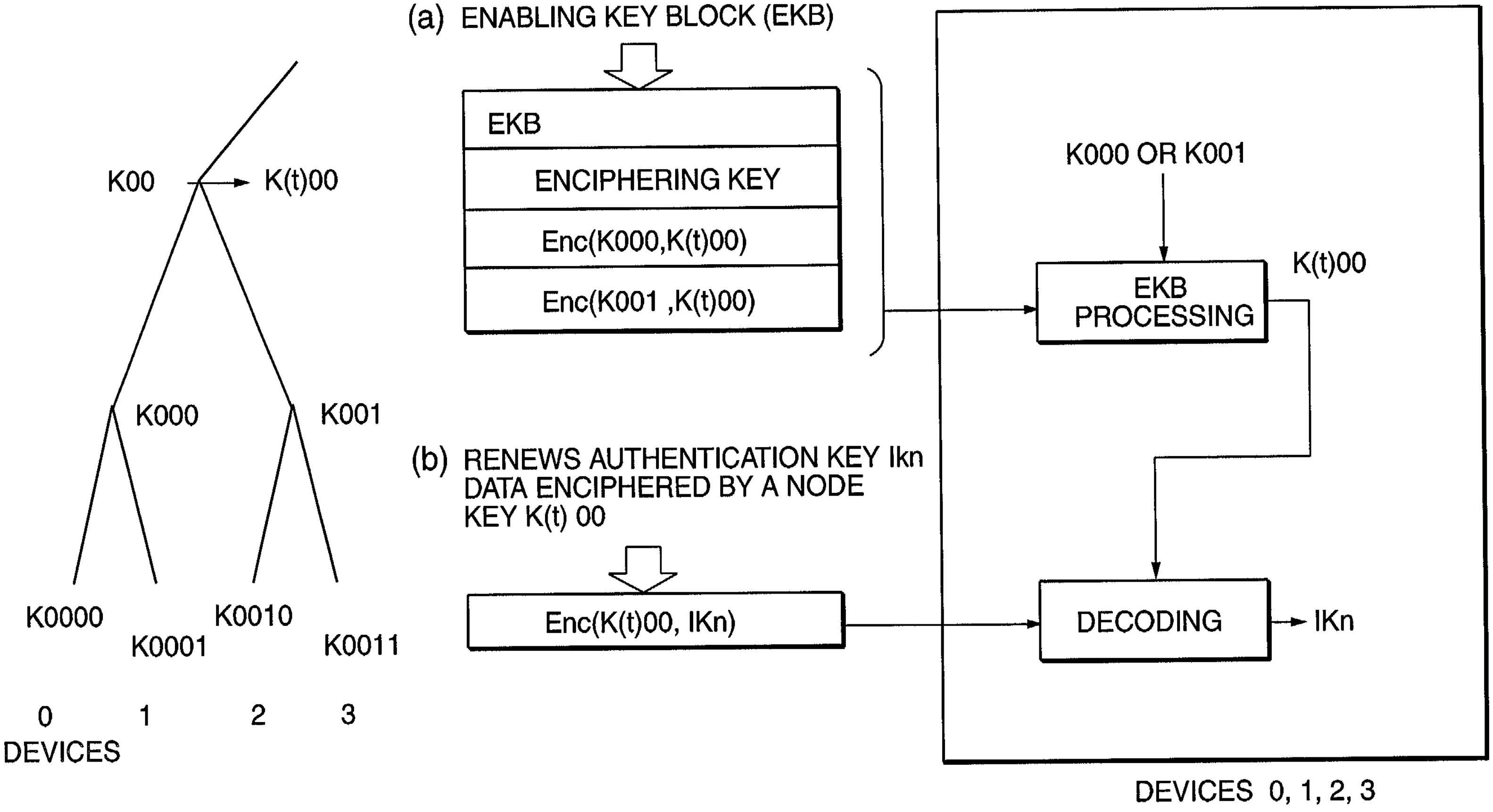


FIG. 33


START
SELECT N OF EKB FILE FORM MORE COUNT NUMBER
S3201
STORE KEK ACQUIRED BY SELECTED EKB PROCESSING INTO RAM
S3202
SELECT DECODING CONTENTS
S3203
IS KEK APPLICABLE TO DECODING CONTENTS STORED IN RAM?
S3204
NO
YES
ACQUIRE CONTENTS KEY Kcon FROM KEK
S3205
IS THERE E(Kstm, Kcon)?
S3206
NO
YES
S3207
ACQUIRE KONTENTS KEY Kcon BASED ON Kstm
S3208
ACQUIRE KONTENTS KEY Kcon FROM EKB
S3209
REPRODUCTION

# FIG. 34



EKB

ENCIPHERING KEY

Enc(K000,K(t)00)

Enc(K001 ,K(t)00)

Enc(K(t)00, IKn)

# FIG. 35

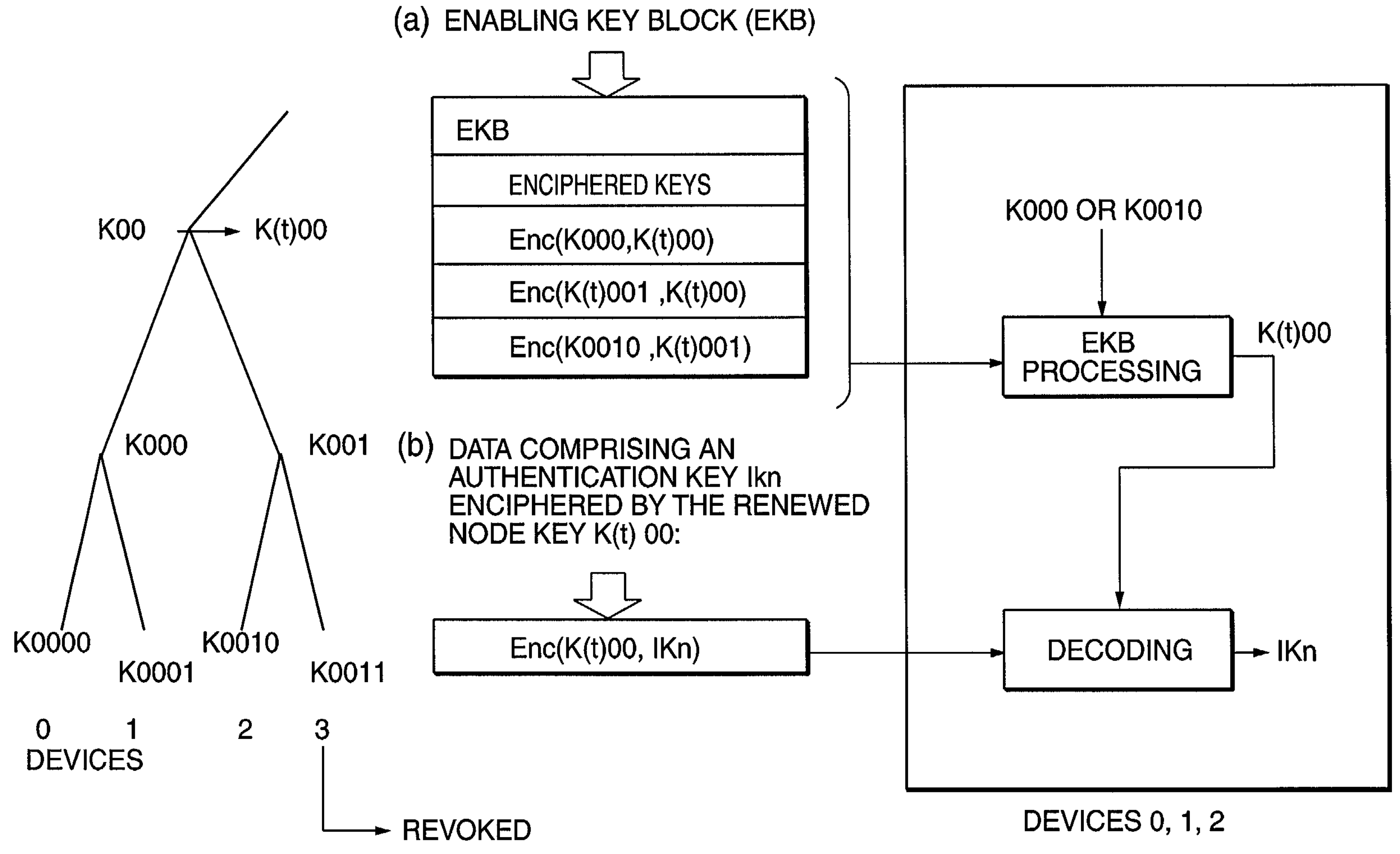
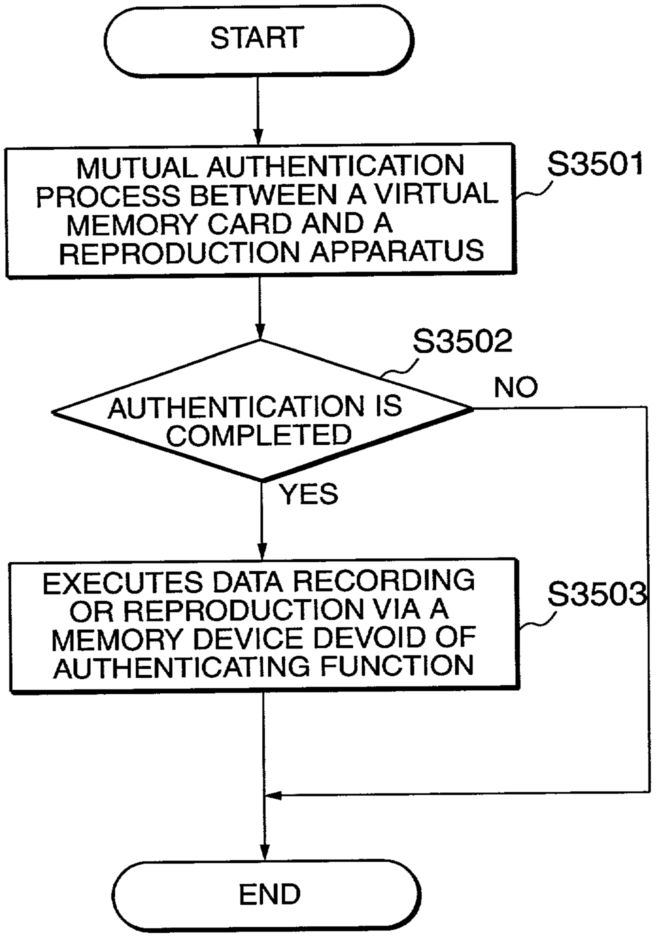
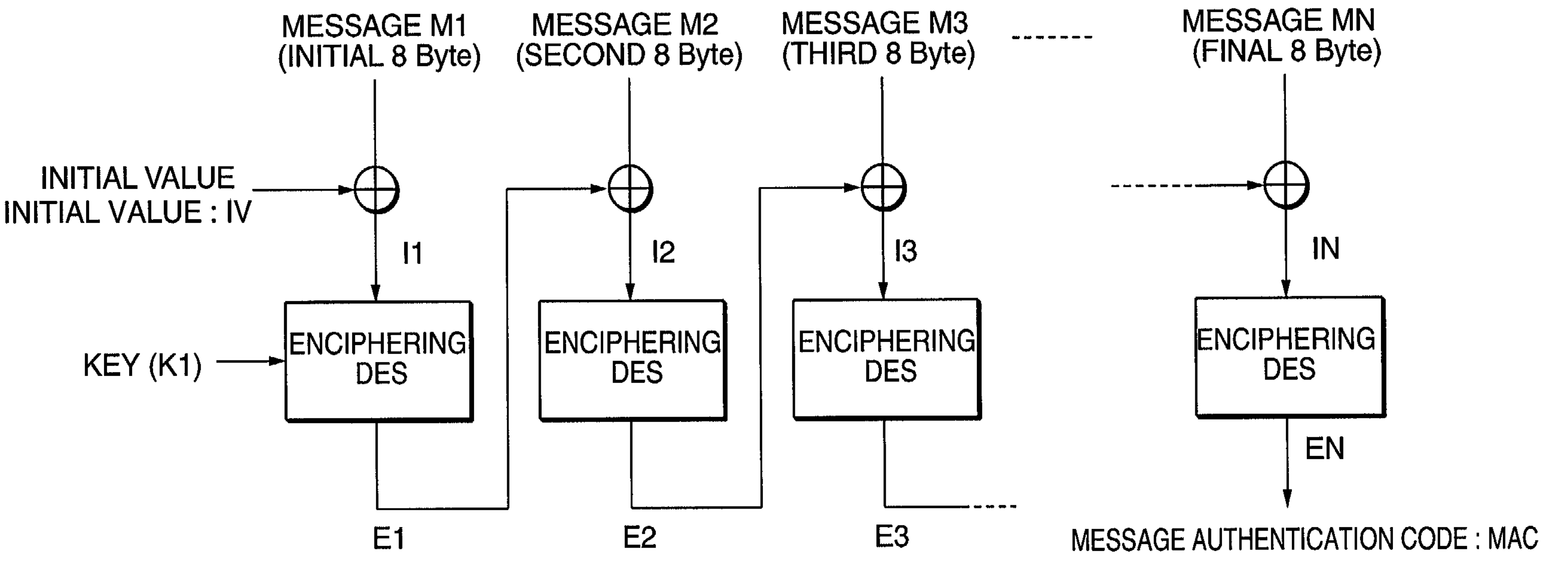

K00
K(t)00
K000
K001
K0000
K0001
K0010
K0011
0
1
2
3
DEVICES
REVOKED
(a) ENABLING KEY BLOCK (EKB)
EKB
ENCIPHERED KEYS
Enc(K000,K(t)00)
Enc(K(t)001 ,K(t)00)
Enc(K0010 ,K(t)001)
(b) DATA COMPRISING AN AUTHENTICATION KEY Ikn ENCIPHERED BY THE RENEWED NODE KEY K(t) 00:
Enc(K(t)00, IKn)
K000 OR K0010
EKB PROCESSING
K(t)00
DECODING
IKn
DEVICES 0, 1, 2

FIG. 36

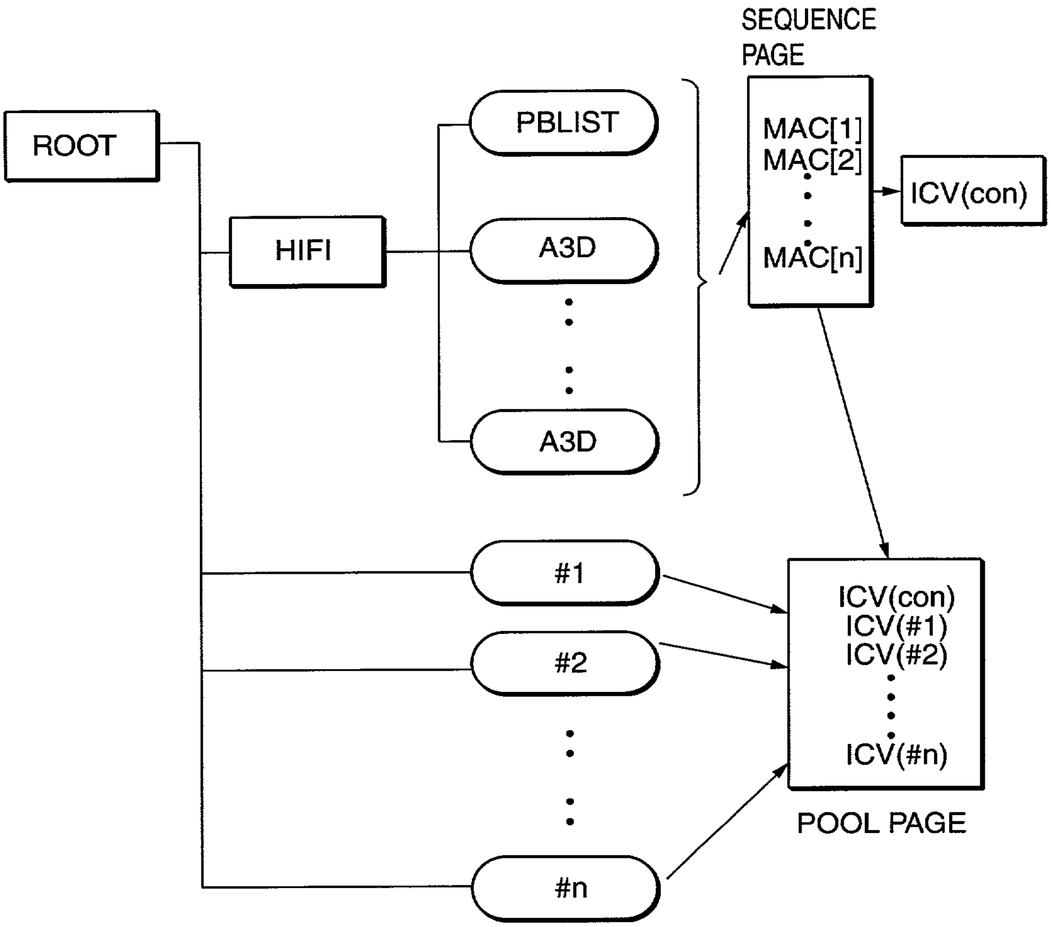

START
MUTUAL AUTHENTICATION PROCESS BETWEEN A VIRTUAL MEMORY CARD AND A REPRODUCTION APPARATUS
S3501
S3502
AUTHENTICATION IS COMPLETED
NO
YES
EXECUTES DATA RECORDING OR REPRODUCTION VIA A MEMORY DEVICE DEVOID OF AUTHENTICATING FUNCTION
S3503
END

# FIG. 37



⊕ EXCLUSIVE OR PROCESS (8 Bytes UNIT)

# FIG. 38


SEQUENCE PAGE
ROOT
HIFI
PBLIST
A3D
A3D
MAC[1]
MAC[2]
MAC[n]
ICV(con)
#1
#2
#n
ICV(con)
ICV(#1)
ICV(#2)
ICV(#n)
POOL PAGE

# FIG. 39

SEQUENCE PAGE FORMAT

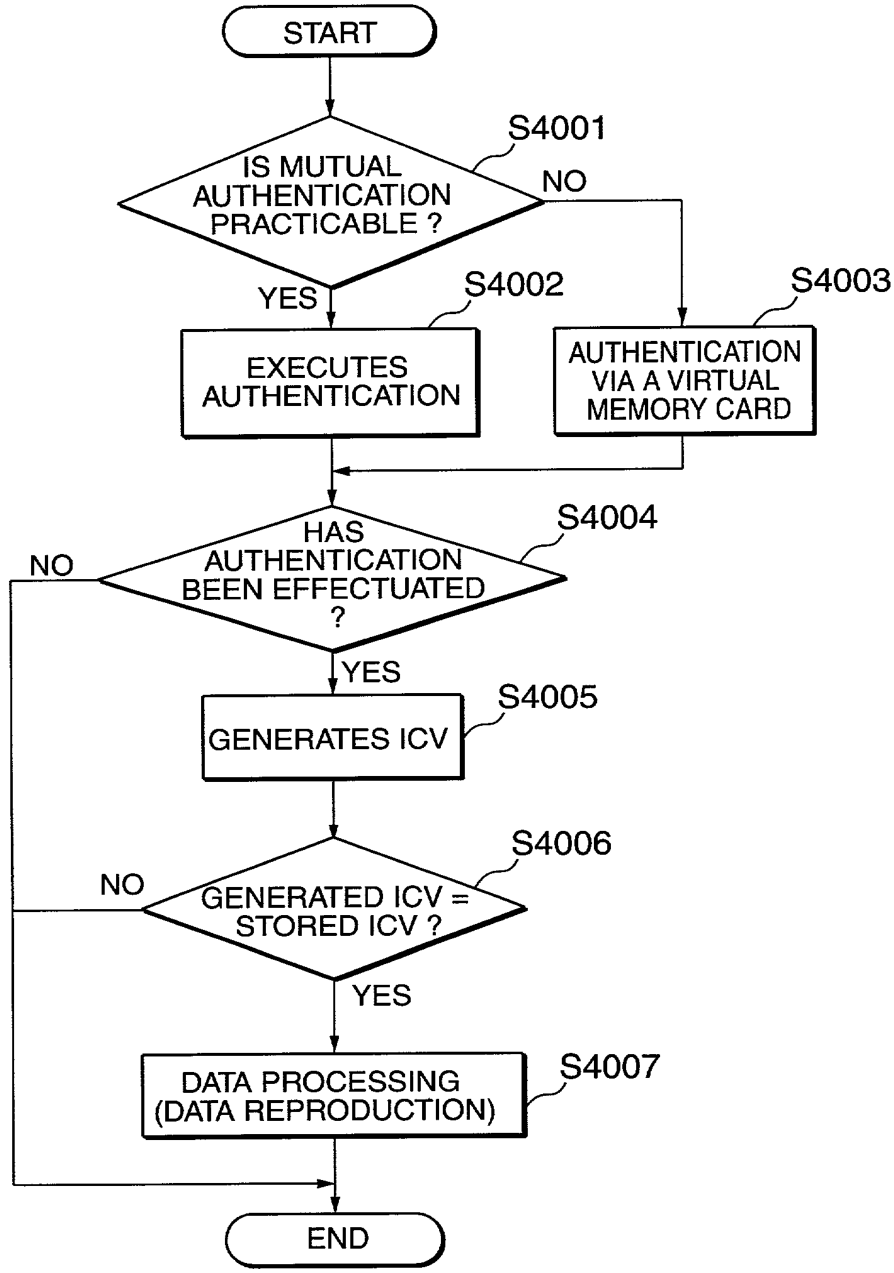
<table>
<tr><th></th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th><th>F</th></tr>
<tr><td>0x0000</td><td colspan="8">E(Kstr, Kcon)</td><td colspan="8">RESERVED</td></tr>
<tr><td>0x0010</td><td colspan="8">ID(Upper)</td><td colspan="8">IO(LOWER)</td></tr>
<tr><td>0x0020</td><td colspan="8">C_MAC[0] (PUBLIST)</td><td colspan="8">C_MAC[1]</td></tr>
<tr><td>0x0030</td><td colspan="8">C_MAC[2]</td><td colspan="8">C_MAC[3]</td></tr>
<tr><td></td><td colspan="16">⋮<br>⋮<br>⋮</td></tr>
<tr><td>0x0FF0</td><td colspan="8">C_MAC[nnn]</td><td colspan="4">RESERVED</td><td colspan="4">REVISION</td></tr>
</table>

# FIG. 40

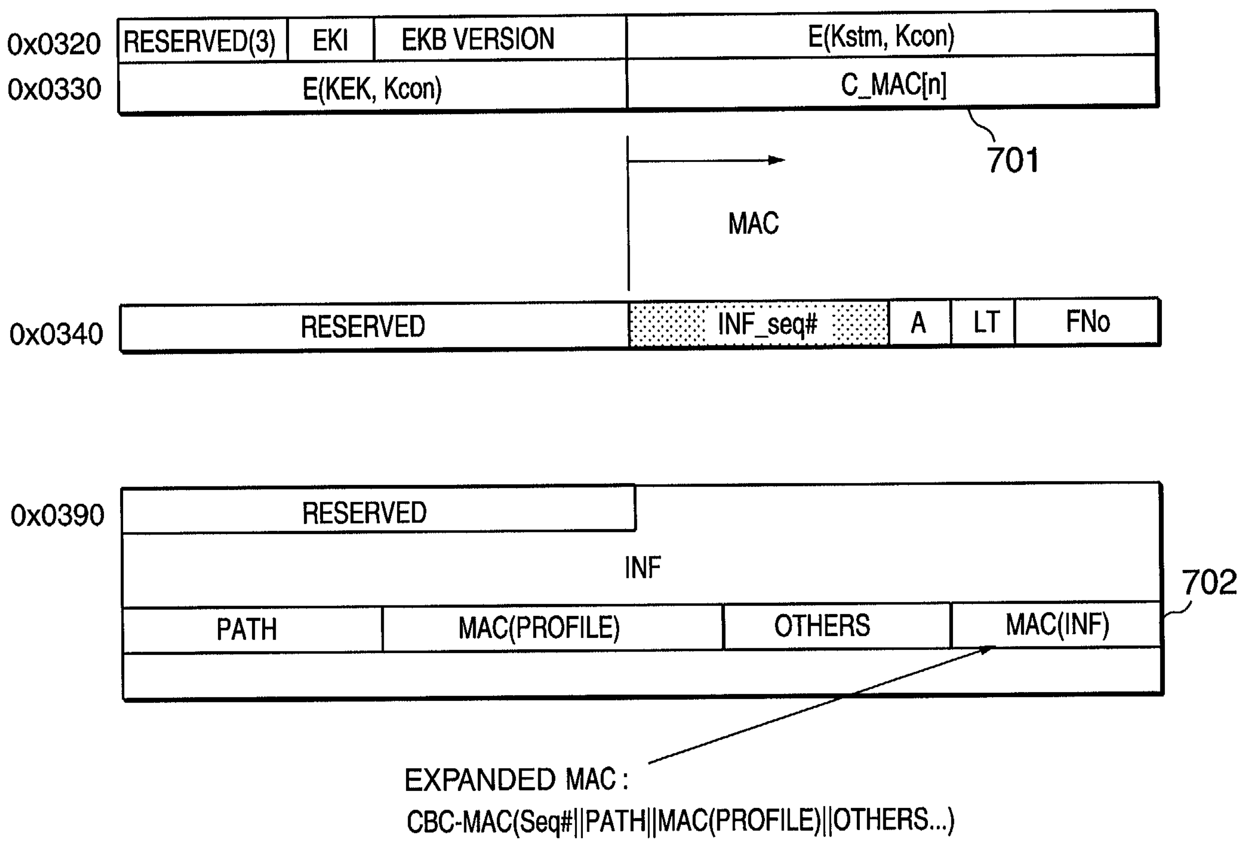
POOL PAGE FORMAT

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x0000 | #0_REVISION | | | | #0_EKB VERSION | | | | #0_E(KEK, Kicv) | | | | | | | |
| 0x0010 | #0_E(KEK, Kicv) | | | | | | | | ICV0 | | | | | | | |
| 0x0020 | #1_REVISION | | | | #1_EKB VERSION | | | | #1_E(KEK, Kicv) | | | | | | | |
| 0x0030 | #1_E (KEK,Kicv) | | | | | | | | ICV1 | | | | | | | |
| | | | | | | | | | ⋮ | | | | | | | |
| 0x01E0 | #15_REVISION | | | | #15_EKB VERSION | | | | #15_E(KEK, Kicv) | | | | | | | |
| 0x01F0 | #15_E (KEK,Kicv) | | | | | | | | ICV15 | | | | | | | |

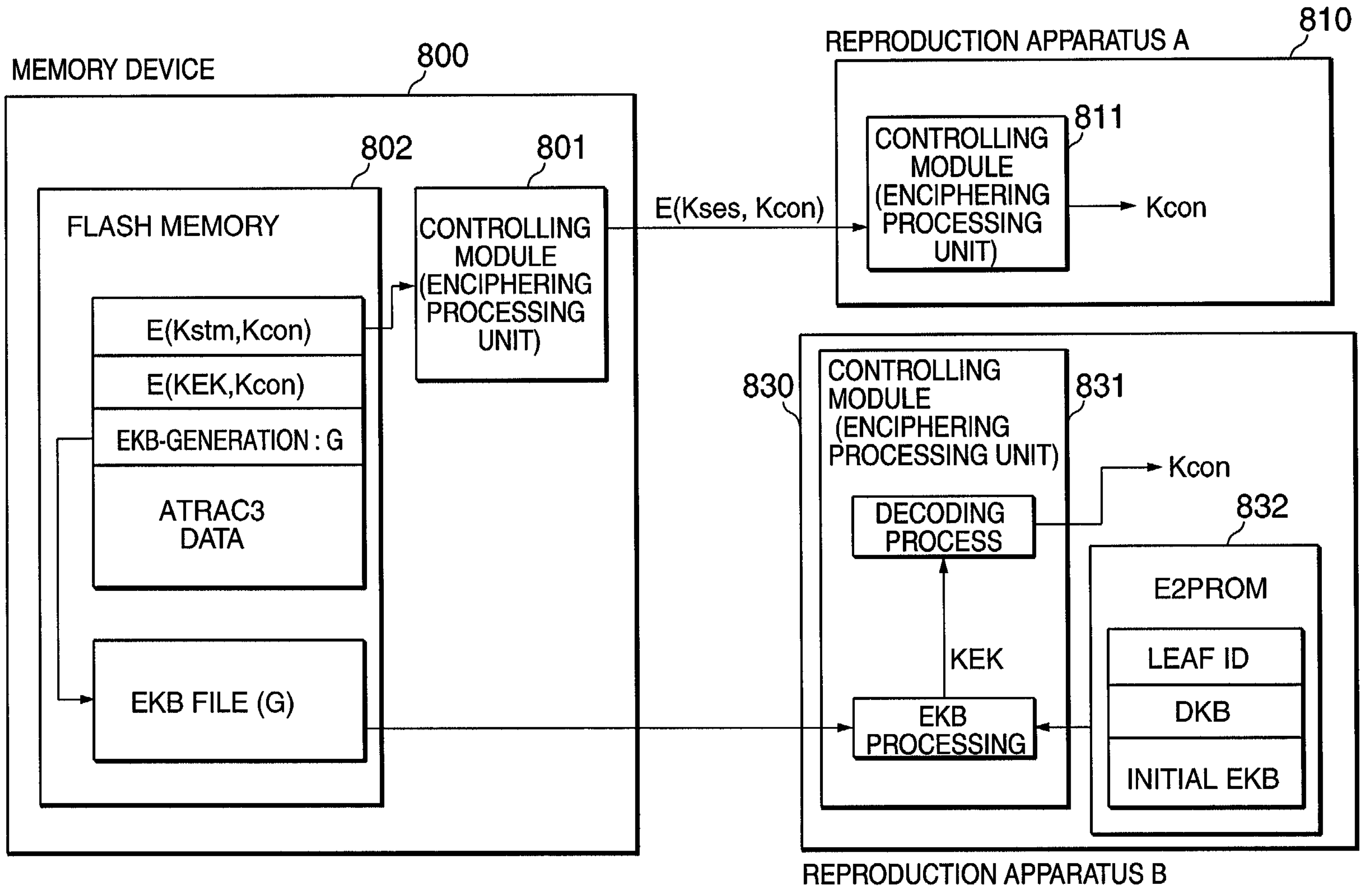
FIG. 41


START
S4001
IS MUTUAL AUTHENTICATION PRACTICABLE ?
NO
YES
S4002
EXECUTES AUTHENTICATION
S4003
AUTHENTICATION VIA A VIRTUAL MEMORY CARD
S4004
HAS AUTHENTICATION BEEN EFFECTUATED ?
NO
YES
S4005
GENERATES ICV
S4006
GENERATED ICV = STORED ICV ?
NO
YES
S4007
DATA PROCESSING (DATA REPRODUCTION)
END

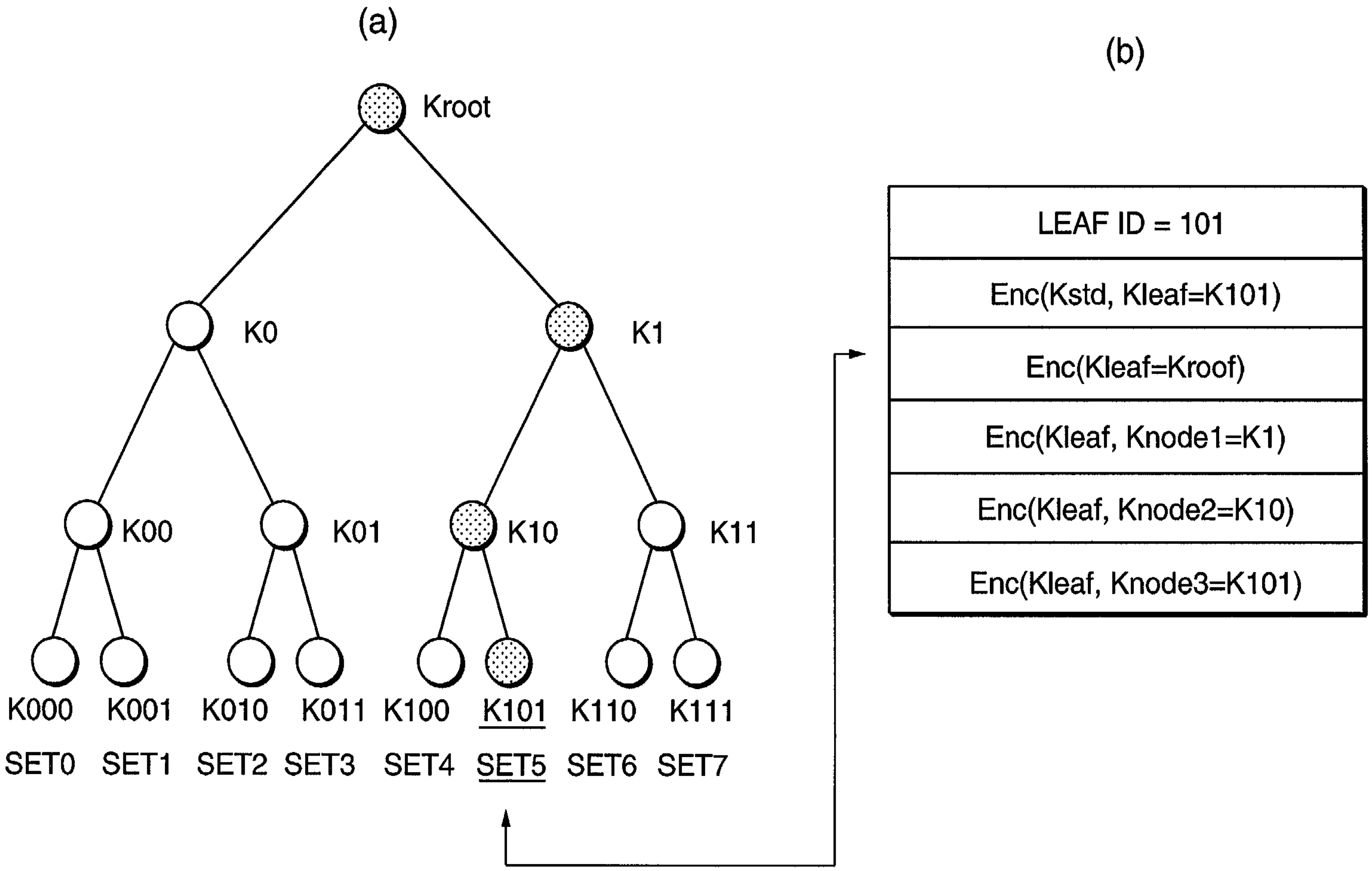
# FIG. 42


0x0320
RESERVED(3)
EKI
EKB VERSION
E(Kstm, Kcon)
0x0330
E(KEK, Kcon)
C_MAC[n]
701
MAC
0x0340
RESERVED
INF_seq#
A
LT
FNo
0x0390
RESERVED
INF
702
PATH
MAC(PROFILE)
OTHERS
MAC(INF)
EXPANDED MAC :
CBC-MAC(Seq#||PATH||MAC(PROFILE)||OTHERS...)

# FIG. 43


MEMORY DEVICE
800
FLASH MEMORY
802
E(Kstm,Kcon)
E(KEK,Kcon)
EKB-GENERATION : G
ATRAC3 DATA
EKB FILE (G)
801
CONTROLLING MODULE (ENCIPHERING PROCESSING UNIT)
E(Kses, Kcon)
REPRODUCTION APPARATUS A
810
811
CONTROLLING MODULE (ENCIPHERING PROCESSING UNIT)
Kcon
830
831
CONTROLLING MODULE (ENCIPHERING PROCESSING UNIT)
DECODING PROCESS
Kcon
KEK
EKB PROCESSING
832
E2PROM
LEAF ID
DKB
INITIAL EKB
REPRODUCTION APPARATUS B

# FIG. 44


(a)
Kroot
K0
K1
K00
K01
K10
K11
K000
K001
K010
K011
K100
K101
K110
K111
SET0
SET1
SET2
SET3
SET4
SET5
SET6
SET7
(b)
LEAF ID = 101
Enc(Kstd, Kleaf=K101)
Enc(Kleaf=Kroof)
Enc(Kleaf, Knode1=K1)
Enc(Kleaf, Knode2=K10)
Enc(Kleaf, Knode3=K101)

# FIG. 45

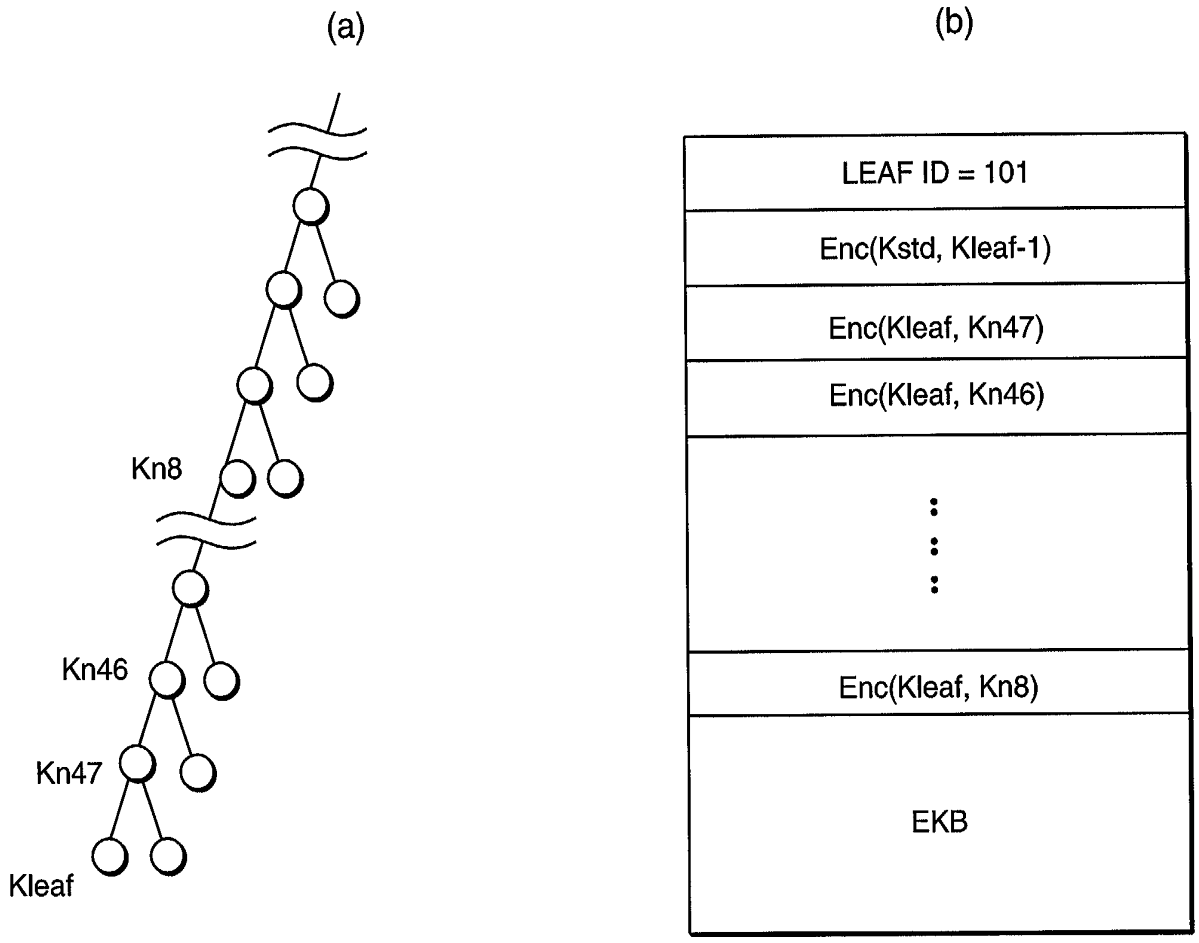
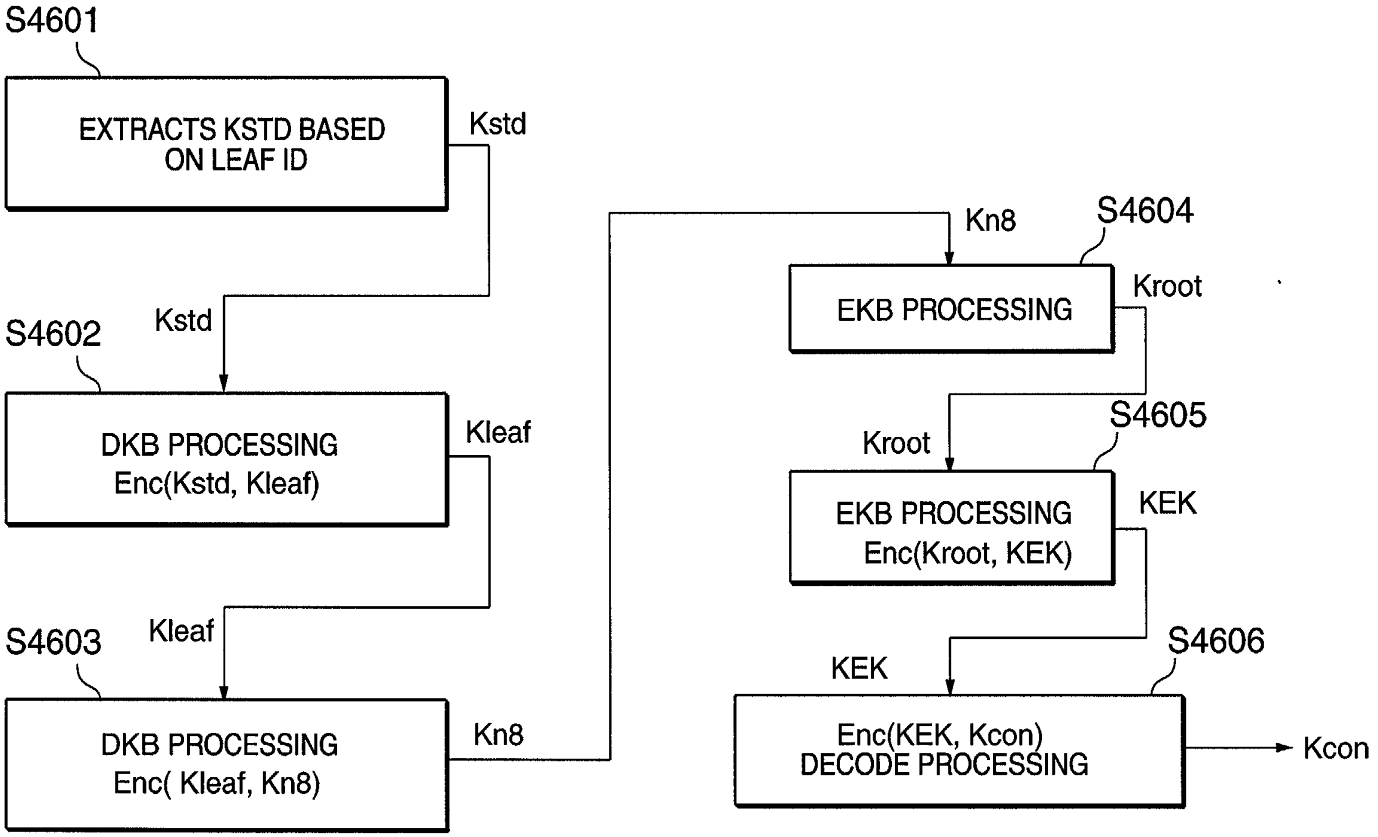

(a)
Kn8
Kn46
Kn47
Kleaf
(b)
LEAF ID = 101
Enc(Kstd, Kleaf-1)
Enc(Kleaf, Kn47)
Enc(Kleaf, Kn46)
⋮
Enc(Kleaf, Kn8)
EKB

FIG. 46


S4601
EXTRACTS KSTD BASED ON LEAF ID
Kstd
Kstd
S4602
DKB PROCESSING Enc(Kstd, Kleaf)
Kleaf
Kleaf
S4603
DKB PROCESSING Enc( Kleaf, Kn8)
Kn8
Kn8
S4604
EKB PROCESSING
Kroot
Kroot
S4605
EKB PROCESSING Enc(Kroot, KEK)
KEK
KEK
S4606
Enc(KEK, Kcon) DECODE PROCESSING
Kcon